

**PERAMALAN HARGA SAHAM MENGGUNAKAN METODE
EXTREME LEARNING MACHINE (ELM) STUDI KASUS SAHAM
BANK MANDIRI**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Muhammad Iqbal Pratama

NIM: 145150201111031



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PERAMALAN HARGA SAHAM MENGGUNAKAN METODE *EXTREME LEARNING MACHINE* (ELM) STUDI KASUS SAHAM BANK MANDIRI

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muhammad Iqbal Pratama
NIM: 145150201111031

Skripsi ini telah diuji dan dinyatakan lulus pada
6 Juni 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Putra Pandu Adikara, S.Kom, M.Kom
NIP: 19850725 200812 1 002

Sigit Adinugroho, S.Kom, M.Sc
NIK: 2016078807011001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

IDENTITAS TIM PENGUJI

- Penguji I (Majelis penguji ujian skripsi)
 - Budi Darma Setiawan, S.Kom, M.Cs
 - NIP. 198410152014041002
- Penguji II
 - Bayu Rahayudi, S.T, M.T
 - NIP. 19740712 200604 1 001



PERNYATAAN ORISINALITAS

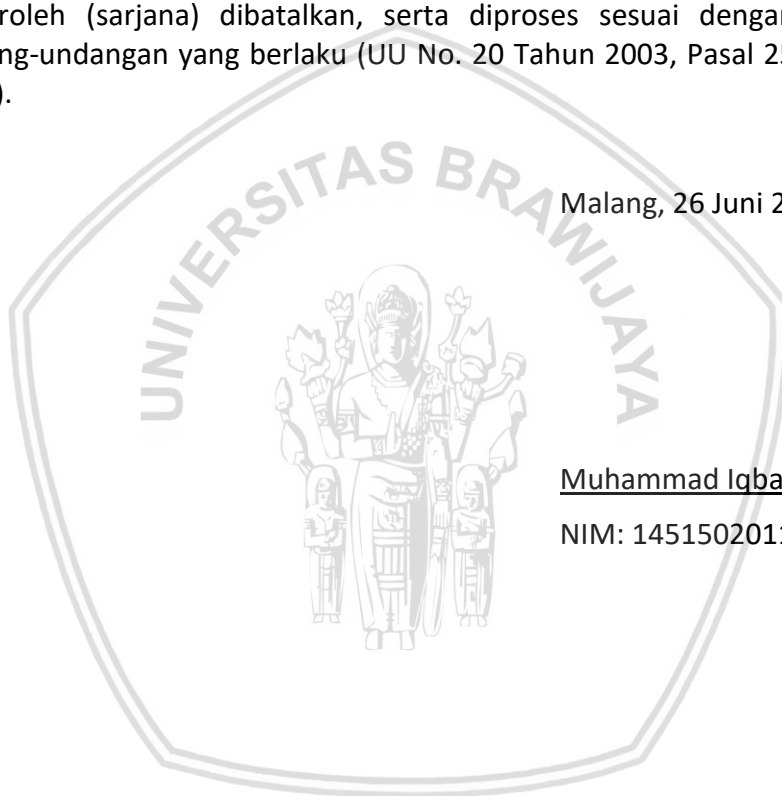
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 26 Juni 2018

Muhammad Iqbal Pratama

NIM: 145150201111031



ABSTRAK

Muhammad Iqbal Pratama, Peramalan Harga Saham Menggunakan Metode *Extreme Learning Machine* (ELM) Studi Kasus Saham Bank Mandiri

Pembimbing: Putra Pandu Adikara, S.Kom, M.Kom dan Sigit Adinugroho, S.Kom, M.Sc

Saham adalah salah satu jenis investasi yang dapat menghasilkan keuntungan terbesar. Salah satu masalah yang ada pada investasi saham adalah sulitnya meramal harga saham sehingga menimbulkan keraguan untuk membeli atau menjual suatu saham. Metode ELM diimplementasikan untuk peramalan harga saham dengan studi kasus saham harian Bank Mandiri. Metode ini memiliki keunggulan berupa waktu pelatihan yang cepat dan nilai *error* yang rendah. Proses yang dilakukan adalah melakukan normalisasi terhadap data saham harian Bank Mandiri, membangkitkan bobot input dan bobot bias, melakukan tahap pelatihan, melakukan tahap pengujian, melakukan denormalisasi terhadap data hasil peramalan, dan melakukan evaluasi model menggunakan *Mean Absolute Percentage Error* (MAPE). Fitur berupa harga *Open*, *High* dan *Low* akan digunakan untuk meramal harga saham harian Bank Mandiri. Dari hasil pengujian, didapatkan nilai MAPE terendah sebesar 1.016% dengan menggunakan fungsi aktivasi sigmoid, jumlah neuron pada hidden layer sebanyak empat buah neuron dan data yang digunakan adalah data satu tahun terakhir.

Kata kunci: Extreme Learning Machine, Saham, MAPE

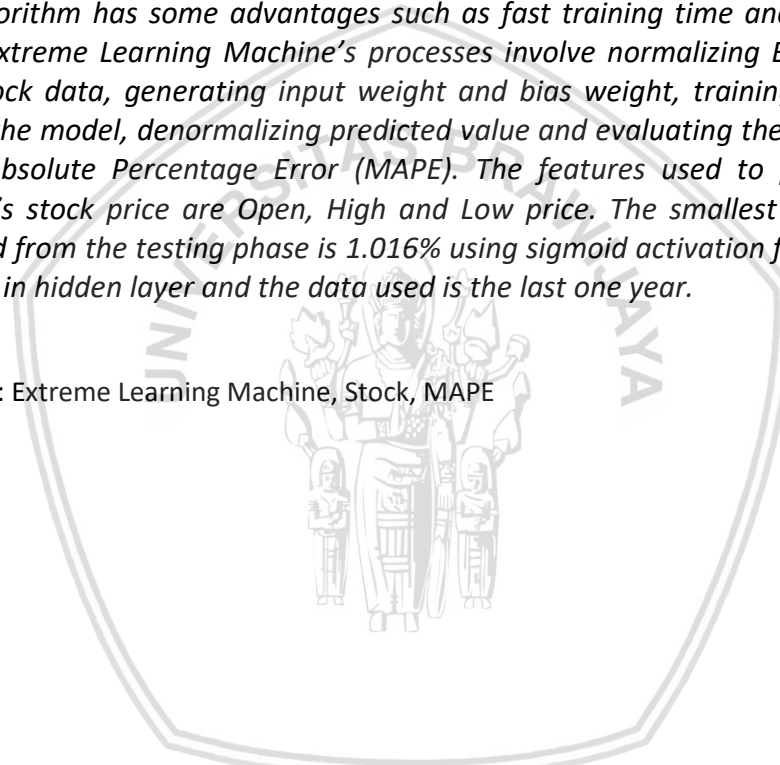
ABSTRACT

Muhammad Iqbal Pratama, Stock Price Prediction Using *Extreme Learning Machine* (ELM) with Bank Mandiri's Stock as Study Case

Supervisor: Putra Pandu Adikara, S.Kom, M.Kom and Sigit Adinugroho, S.Kom, M.Sc

Stock investment is one of the most profitable type of investment. One of the biggest problem in stock investing is the difficultness to predict a stock price and it led to doubt whether to buy or sell a stock. Extreme Learning Machine is implemented to predict a stock price using Bank Mandiri's stock as a case study. This algorithm has some advantages such as fast training time and small error value. Extreme Learning Machine's processes involve normalizing Bank Mandiri daily stock data, generating input weight and bias weight, training the model, testing the model, denormalizing predicted value and evaluating the model using Mean Absolute Percentage Error (MAPE). The features used to predict Bank Mandiri's stock price are Open, High and Low price. The smallest MAPE value obtained from the testing phase is 1.016% using sigmoid activation function, four neurons in hidden layer and the data used is the last one year.

Keyword: Extreme Learning Machine, Stock, MAPE



KATA PENGANTAR

Puji dan syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat yang telah diberikan-Nya sehingga skripsi yang berjudul “Peramalan Harga Saham Menggunakan Metode *Extreme Learning Machine* (ELM) Studi Kasus Saham Bank Mandiri” ini dapat penulis selesaikan tepat waktu.

Tujuan dari penulisan skripsi ini adalah sebagai salah satu syarat untuk memperoleh gelar sarjana komputer dari Fakultas Ilmu Komputer Universitas Brawijaya Malang.

Pengerjaan skripsi ini tentunya telah melibatkan banyak pihak yang sangat membantu dalam berbagai hal. Oleh karena itu, penulis ingin mengucapkan rasa terimakasih yang sebesar-besarnya kepada:

1. Bapak Putra Pandu Adikara, S.Kom, M.Kom dan Bapak Sigit Adinugroho, S.Kom, M.Sc selaku Pembimbing skripsi yang telah membimbing penulis dengan sabar dalam menyusun skripsi hingga selesai.
2. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang yang telah menyetujui permohonan penyusunan skripsi.
4. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang yang telah memberikan ijin penelitian.
5. Orang tua penulis tercinta yang senantiasa memberikan doa dan dukungan kepada penulis sehingga penulis dapat menyelesaikan skripsi ini tepat waktu.
6. Vioentina Yofiani, pacar penulis yang selalu mendoakan dan mendukung penulis untuk menyelesaikan skripsi ini dari Bekasi.
7. Sendi, Danang, Adit, Josua, Bambang, Adhy dan Kamal selaku teman terdekat penulis selama di Malang yang bersama-sama menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya Malang.
8. Rekan-rekan BIOS Exalt yang sudah memberikan pengalaman-pengalaman yang indah selama satu periode.
9. Teman-teman kelas Informatika C tahun 2014 yang sudah membantu penulis beradaptasi dengan kehidupan di Kota Malang.
10. Semua pihak yang banyak memberi bantuan yang tidak bisa disebutkan satu persatu.

Penulis menyadari bahwa penyusunan naskah skripsi ini masih jauh dari kata sempurna, sehingga saran dan kritik yang membangun sangat penulis harapkan.

Akhir kata, semoga skripsi ini dapat bermanfaat bagi semua pihak yang menggunakannya.

Malang, 26 Juni 2018

Penulis

lbaypratama20@gmail.com



DAFTAR ISI

PENGESAHAN	ii
IDENTITAS TIM PENGUJI	iii
PERNYATAAN ORISINALITAS	iv
ABSTRAK.....	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
DAFTAR ISI	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Saham	6
2.3 Peramalan	7
2.3.1 Data <i>Time Series</i>	7
2.4 Jaringan Saraf Tiruan	7
2.4.1 Algoritme <i>Extreme Learning Machine</i>	8
2.4.2 Tahap Pelatihan <i>Extreme Learning Machine</i>	8
2.4.3 Tahap Pengujian <i>Extreme Machine Learning</i>	9
2.5 Evaluasi	10
2.6 <i>Cross Validation</i>	10
BAB 3 METODOLOGI PENELITIAN	12
3.1 Tipe Penelitian	12

3.2 Strategi dan Rancangan Penelitian	12
3.2.1 Metode Secara Umum	12
3.2.2 Objek Penelitian	13
3.2.3 Lokasi Penelitian.....	14
3.2.4 Metode Pengumpulan Data	14
3.2.5 Peralatan Pendukung	14
3.3 Jadwal Kegiatan	14
BAB 4 PERANCANGAN DAN IMPLEMENTASI	15
4.1 Formulasi Permasalahan.....	15
4.2 Penyelesaian Permasalahan	15
4.2.1 Normalisasi Data	17
4.2.2 Pembangkitan Bobot Input dan Bobot Bias	17
4.2.3 Tahap Pelatihan.....	19
4.2.4 Tahap Pengujian	25
4.2.5 Denormalisasi	27
4.2.6 Evaluasi.....	28
4.3 Perhitungan Manual	29
4.3.1 Perhitungan Manual Proses Pelatihan	32
4.3.2 Perhitungan Manual Proses Pengujian	34
4.3.3 Proses Denormalisasi	35
4.3.4 Perhitungan Nilai MAPE	35
4.4 Perancangan Antarmuka	35
4.5 Pengujian Metode.....	36
4.5.1 Pengujian Fungsi Aktivasi	37
4.5.2 Pengujian Kombinasi Jumlah Neuron dan Variasi Jumlah Data	37
4.6 Implementasi Algoritme	38
4.6.1 Implementasi Normalisasi Data	38
4.6.2 Implementasi Pembangkitan Bobot Input dan Bobot Bias.....	38
4.6.3 Implementasi Pembangkitan Matriks Inisialisasi	39
4.6.4 Implementasi Fungsi Aktivasi.....	39
4.6.5 Implementasi <i>Pseudo-Inverse</i>	40
4.6.6 Implementasi Menghitung Nilai β	40

4.6.7 Implementasi Proses Menghitung Hasil Peramalan	40
4.6.8 Implementasi Denormalisasi Hasil Peramalan.....	41
4.6.9 Implementasi Evaluasi Hasil Peramalan	41
4.7 Implementasi Antarmuka	42
BAB 5 PENGUJIAN DAN ANALISIS.....	43
5.1 Pengujian Fungsi Aktivasi.....	43
5.2 Pengujian Kombinasi Jumlah Neuron dan Variasi Jumlah Data	43
5.3 Pengujian Dengan Data Uji yang Sama Pada Setiap Skenario	45
5.4 Analisis Hasil Pengujian.....	46
5.4.1 Analisis Hasil Pengujian Fungsi Aktivasi	46
5.4.2 Analisis Hasil Pengujian Kombinasi Jumlah Neuron dan Variasi Jumlah Data.....	46
5.4.3 Analisis Hasil Pengujian Dengan Data Uji yang Sama Pada Setiap Skenario.....	48
5.3 <i>Cross Validation Data Time Series</i>	48
BAB 6 KESIMPULAN DAN SARAN	50
6.1 Kesimpulan.....	50
6.2 Saran	50
DAFTAR PUSTAKA.....	51
LAMPIRAN A SAMPEL DATA SAHAM HARIAN BANK MANDIRI	53

DAFTAR TABEL

Tabel 2.1 Tabel Kajian Pustaka	6
Tabel 2.2 Data saham harian	7
Tabel 3.1 Rencana Jadwal Kegiatan	14
Tabel 4.1 Data saham harian Bank Mandiri	29
Tabel 4.2 Data latih	30
Tabel 4.3 Data uji	30
Tabel 4.4 Nilai maksimum dan minimum fitur <i>Open</i>	30
Tabel 4.5 Normalisasi	31
Tabel 4.6 Matriks bobot	31
Tabel 4.7 Matriks transpose bobot	31
Tabel 4.8 Matriks bobot bias	32
Tabel 4.9 Matriks inisialisasi	32
Tabel 4.10 Matriks fungsi aktivasi	32
Tabel 4.11 Matriks transpose fungsi aktivasi	33
Tabel 4.12 Matriks HTH	33
Tabel 4.13 Matriks Inverse	33
Tabel 4.14 Matriks H +	34
Tabel 4.15 Nilai Beta estimasi	34
Tabel 4.16 Matriks inisialisasi data uji	34
Tabel 4.17 Matriks aktivasi data uji	34
Tabel 4.18 Vektor hasil peramalan	35
Tabel 4.19 Vektor denormalisasi	35
Tabel 4.20 Perancangan pengujian fungsi aktivasi	37
Tabel 4.21 Perancangan pengujian kombinasi jumlah neuron dan variasi jumlah data	37
Tabel 5.1 Pengujian fungsi aktivasi sigmoid, sin dan tanh	43
Tabel 5.2 Pengujian jumlah neuron pada data 5 tahun terakhir	43
Tabel 5.3 Pengujian jumlah neuron pada data 3 tahun terakhir	44
Tabel 5.4 Pengujian jumlah neuron pada data 1 tahun terakhir	44
Tabel 5.5 Pengujian dengan data uji yang sama pada setiap scenario	45

Tabel 5.6 Tabel <i>Cross Validation</i>	49
---	----



DAFTAR GAMBAR

Gambar 2.1 Arsitektur jaringan saraf tiruan	8
Gambar 2.2 Proses <i>k-Fold</i> pada model peramalan <i>time-series</i>	10
Gambar 3.1 Diagram alir gambaran umum sistem	13
Gambar 4.1 Diagram alir metode <i>Extreme Learning Machine</i>	16
Gambar 4.2 Diagram alir normalisasi data	17
Gambar 4.3 Diagram alir pembangkitan matriks bobot input	18
Gambar 4.4 Diagram alir pembangkitan matriks bobot bias	19
Gambar 4.5 Diagram alir proses pelatihan	20
Gambar 4.6 Diagram alir pembangkitan matriks inisialisasi	22
Gambar 4.7 Diagram alir fungsi aktivasi	23
Gambar 4.8 Diagram alir <i>pseudo-inverse</i>	24
Gambar 4.9 Diagram alir menghitung nilai β	25
Gambar 4.10 Diagram alir proses pengujian	26
Gambar 4.11 Diagram alir proses menghitung hasil peramalan	27
Gambar 4.12 Diagram alir proses denormalisasi	28
Gambar 4.13 Diagram alir proses evaluasi	29
Gambar 4.14 Perancangan antarmuka	36
Gambar 4.15 Kode program normalisasi data	38
Gambar 4.16 Kode program pembangkitan bobot input	39
Gambar 4.17 Kode program pembangkitan bobot bias	39
Gambar 4.18 Kode program pembangkitan matriks inisialisasi	39
Gambar 4.19 Kode program fungsi aktivasi	40
Gambar 4.20 Kode program <i>Pseudo-Inverse</i>	40
Gambar 4.21 Kode program menghitung nilai β	40
Gambar 4.22 Kode program proses menghitung hasil peramalan	41
Gambar 4.23 Kode program denormalisasi hasil peramalan	41
Gambar 4.24 Kode program evaluasi hasil peramalan	42
Gambar 4.25 Gambar implementasi antarmuka	42
Gambar 5.1 Kombinasi jumlah neuron dan variasi jumlah data yang digunakan	45
Gambar 5.2 Grafik perbandingan nilai real dan nilai hasil peramalan	46

Gambar 5.3 Pergerakan harga saham harian Bank Mandiri 5 tahun terakhir.....	47
Gambar 5.4 Pergerakan harga saham harian Bank Mandiri 3 tahun terakhir.....	47
Gambar 5.5 Pergerakan harga saham harian Bank Mandiri 1 tahun terakhir.....	48



DAFTAR LAMPIRAN

LAMPIRAN A SAMPEL DATA SAHAM HARIAN BANK MANDIRI	53
--	----



BAB 1 PENDAHULUAN

1.1 Latar belakang

Investasi saham adalah salah satu bentuk usaha menambah pendapatan pasif yang paling baik, karena saham merupakan bentuk investasi yang dapat memberikan keuntungan paling tinggi. Investasi saham akan sangat berguna apabila masyarakat memiliki uang simpanan yang tidak berisiko apabila diinvestasikan. Investasi saham nantinya dapat membantu masyarakat untuk meningkatkan nilai uang yang dimilikinya, sehingga uang tersebut dapat digunakan untuk masa tua ataupun keperluan lainnya dimasa depan. Selain itu, investasi saham juga dapat berguna untuk melawan laju inflasi. Harga kebutuhan masyarakat akan terus naik secara terus menerus sehingga menyebabkan nilai mata uang menurun. Hal ini dapat diatasi dengan menginvestasikan uang dalam bentuk saham.

Ada dua tipe investasi saham yang banyak dilakukan investor aktif, yaitu investasi jangka pendek dan investasi jangka panjang. Pada investasi jangka pendek, biasanya investor saham akan melakukan transaksi jual beli saham yang dimilikinya dalam kurun waktu kurang dari satu tahun. Sedangkan pada investasi jangka panjang, investor saham cenderung akan menahan saham yang dimilikinya lebih dari satu tahun. Namun kedua tipe investasi ini memiliki masalah yang sama, yaitu sulitnya meramal pergerakan harga saham harian (Wang, Wu, 2010). Masyarakat awam yang baru memasuki dunia investasi tidak memiliki pandangan yang jelas kapan mereka harus membeli atau menjual saham yang mereka miliki. Mereka cenderung tertipu dengan pergerakan saham yang terus naik namun tiba-tiba turun secara drastis sehingga mengakibatkan kerugian yang cukup besar. Begitu juga sebaliknya, mereka cenderung takut untuk membeli suatu saham apabila saham tersebut cenderung bergerak menurun, namun sebenarnya saham ini sudah akan masuk ke fase dimana saham tersebut akan bergerak naik secara cepat. Apabila tidak cermat dalam mengamati pergerakan harga saham, investasi saham tentu akan menimbulkan kerugian.

Penelitian tentang peramalan harga saham sudah pernah dilakukan sebelumnya, salah satunya adalah penelitian yang dilakukan Zeng-Min dan Chong. Pada penelitian tersebut, Zeng-Min menggunakan metode *Support Vector Regression* (SVR) untuk memprediksi index saham Nikkei 225. Penelitian tersebut kemudian dievaluasi menggunakan *Mean Absolute Percentage Error* (MAPE). Hasilnya, nilai MAPE yang didapatkan dari implementasi metode SVR untuk peramalan saham adalah sebesar 1.4676. Nilai MAPE tersebut cukup baik melihat nilai yang dihasilkan berada di bawah 10%. Namun metode SVR sendiri memiliki kelemahan, salah satunya adalah masalah kecepatan pelatihan. Masalah ini akan berdampak buruk apabila data yang digunakan untuk peramalan nantinya berjumlah besar.

Beberapa algoritme peramalan sudah banyak diteliti, salah satu yang memiliki hasil *error* yang rendah dan waktu pelatihan yang cepat adalah algoritme *Extreme*

Learning Machine (ELM) yang merupakan sebuah algoritme jaringan saraf tiruan. Dalam penelitiannya, Guang-Bin Huang mengajukan ELM sebagai algoritme untuk regresi dan klasifikasi multiklas. Pada penelitian tersebut, Huang membandingkan performa ELM dengan algoritme regresi lainnya yaitu SVR dan LSSVR pada beberapa kasus. Hasilnya ELM memiliki nilai RMSE yang mengungguli SVR dan LSSVR pada beberapa kasus, ELM juga memiliki waktu pelatihan yang lebih cepat dibandingkan kedua algoritme tersebut.

ELM dan SVR kembali dibandingkan dalam penelitian lainnya mengenai peramalan tingkat penyerapan sumur (Cheng, Cai, Pan, 2009). Data yang digunakan untuk peramalan adalah data sampel yang diambil dari sumur minyak di Cina. Jumlah neuron pada hidden layer ELM diinisialisasi sebanyak 30 buah dengan peningkatan sebanyak 20 buah pada setiap percobaan, dan fungsi aktivasi yang digunakan meliputi *Sin*, *Sigmoid*, *Hardlim*, *Triangular basis* dan *Radial basis*. Setelah percobaan dilakukan, diketahui fungsi aktivasi *Hardlim* dan *Triangular basis* menghasilkan nilai error yang tinggi sehingga kedua fungsi tersebut dicoret. Setelah pengujian dilakukan, didapatkan bahwa ELM mengungguli SVR dalam hal kecepatan waktu pelatihan, nilai *error* pelatihan dan kecepatan waktu pengujian. Sementara itu, nilai *error* pada tahap pengujian ELM juga hanya memiliki selisih sebesar 0.0011 satuan nilai lebih kecil dibandingkan SVR.

Dalam penelitian lainnya mengenai peramalan harga listrik, algoritme ELM diterapkan pada data berbentuk *time series*. Algoritme ELM kemudian dibandingkan dengan algoritme *Simple Moving Average* (SMA), hasilnya nilai MAPE yang dihasilkan oleh ELM lebih baik dibandingkan SMA (Tee, et al., 2017). Hal ini dapat terjadi karena sebuah jaringan saraf tiruan dapat menerima jumlah masukan lebih dari satu, sedangkan SMA hanya mempertimbangkan satu buah variabel masukan. Dalam penelitian ini juga disebutkan bahwa ELM akan menghasilkan peramalan yang lebih akurat dengan data yang berjumlah lebih besar.

Berdasarkan paparan di atas, algoritme ELM akan digunakan untuk peramalan harga saham harian karena algoritme ELM memiliki nilai error yang baik untuk sebuah algoritme regresi dan algoritme ini juga memiliki waktu pelatihan yang cepat, sehingga apabila data saham harus diolah secara *real-time*, algoritme ini tetap akan memberikan hasil yang baik dalam waktu yang relatif lebih cepat.

1.2 Rumusan masalah

Dari latar belakang yang sudah diuraikan di atas, didapatkan rumusan masalah sebagai berikut:

1. Bagaimana hasil perhitungan nilai *error* peramalan harga saham menggunakan metode ELM?
2. Bagaimana kombinasi parameter yang menghasilkan nilai *error* terendah?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah dijabarkan, maka dapat dirumuskan tujuan penelitian sebagai berikut:

1. Untuk mengetahui nilai *error* peramalan harga saham harian menggunakan metode ELM.
2. Untuk mengetahui kombinasi parameter ELM yang menghasilkan nilai *error* terendah.

1.4 Manfaat

Manfaat penelitian ini adalah agar masyarakat yang berinvestasi saham dapat memiliki pandangan mengenai tren pergerakan harga saham ke depan dengan melihat dari harga masa sebelumnya. Manfaat lain yang dapat diambil dari penelitian ini adalah mengetahui apakah metode ELM baik untuk peramalan harga saham.

1.5 Batasan masalah

Agar penelitian memiliki fokus yang jelas, maka dirumuskan batasan masalah sebagai berikut:

1. Data yang digunakan adalah data saham harian Bank Mandiri pada rentang waktu 16 Maret 2012 – 29 Desember 2017 yang didapatkan dari investing.com dengan parameter *Price, High, Open, Low*.
2. Pengujian dilakukan dengan menghitung nilai *Mean Absolute Percentage Error* (MAPE).

1.6 Sistematika pembahasan

Sistematika penulisan penelitian ini secara garis besar akan diuraikan sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini menjelaskan latar belakang dari masalah yang diangkat, solusi yang ditawarkan, tujuan, manfaat, batasan masalah serta sistematika pembahasan.

BAB 2 KAJIAN TEORI

Bab ini menjelaskan teori-teori yang digunakan sebagai landasan dalam penelitian ini. Teori-teori tersebut diperoleh dari beberapa buku literatur, jurnal, serta internet.

BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan metodologi yang akan digunakan dalam penelitian ini. Metodologi penelitian yang dijelaskan meliputi tipe penelitian, metode secara umum, objek penelitian, lokasi penelitian, metode pengumpulan data, peralatan pendukung, serta jadwal kegiatan.

BAB 4 PERANCANGAN DAN IMPLEMENTASI

Bab ini menjelaskan tahap perancangan pengujian, perancangan evaluasi algoritme yang diajukan serta *source code* implementasi algoritme yang diajukan untuk penelitian ini.

BAB 5 PENGUJIAN DAN ANALISIS

Bab ini berisi pembahasan serta analisis hasil pengujian algoritme yang diajukan.

BAB 6 KESIMPULAN DAN SARAN

Bab ini menjabarkan kesimpulan dan saran yang didapat dari penelitian ini.



BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan tentang kajian pustaka dan teori pendukung untuk penelitian ini. Pada bab ini, penulis akan membahas penelitian serupa yang telah dilakukan sebelumnya serta membahas teori seputar saham dan peramalan beserta algoritmenya.

2.1 Kajian Pustaka

Kajian pustaka yang akan dibahas adalah tentang peramalan menggunakan algoritme ELM. Penelitian-penelitian yang digunakan sebagai referensi dapat dilihat pada Tabel 2.1.

Penelitian pertama yang dilakukan oleh Huang, et al., (2012) membahas tentang perbandingan algoritme ELM, Support Vector Machine (SVM) dan Least Square Support Vector Machine (LS-SVM). Penelitian ini menunjukkan beberapa hal, yang pertama adalah ELM memiliki kendala optimasi yang lebih ringan dibandingkan dengan SVM dan LS-SVM. Kedua, ELM dapat mencapai solusi suboptimal dengan kompleksitas yang lebih rendah. Yang terakhir, berdasarkan hasil simulasi, ELM cenderung memiliki skalabilitas yang lebih baik dan dapat mencapai performa generalisasi yang mirip dalam waktu pembelajaran yang lebih cepat dibandingkan SVM dan LS-SVM.

Penelitian kedua yang dilakukan oleh Lazarevska (2016) membahas tentang prediksi kecepatan angin menggunakan algoritme ELM. Model yang ada pada penelitian ini dibangun berdasarkan data meteorologi yang tersedia pada Mauna Loa (MOA), Hawaii, Amerika Serikat pada tahun 2015. Beberapa parameter yang termasuk dalam data ini adalah arah angin, kecepatan angin, faktor kemantapan angin, tekanan barometrik, temperatur pada ketinggian 2 meter dan 10 meter, temperatur pada *tower* tertinggi, kelembapan relatif dan intensitas presipitasi. Model ELM untuk prediksi kecepatan angin pada penelitian ini dibandingkan dengan model lain yang didasari oleh teori pembelajaran statistik, logika *fuzzy* dan logika *neuro-fuzzy*. Hasilnya adalah model ELM memiliki kinerja terbaik dibandingkan model-model lain.

Penelitian ketiga yang dilakukan oleh Chazal, et al., (2015) membahas tentang perbandingan algoritme ELM dengan *Backpropagation*. Penelitian ini menunjukkan bahwa dalam hal *error-rate*, *backpropagation* masih mengungguli ELM. Namun dalam hal waktu pelatihan, ELM sangat jauh mengungguli *backpropagation* (hingga 145x lebih cepat).

Penelitian terakhir yang dilakukan oleh Giutsi, et al., (2017) membahas tentang penerapan algoritme ELM untuk data *time series*. Dalam penelitian ini, data yang digunakan adalah data penjualan harian di Kober Mie Setan cabang Soekarno Hatta, Malang. Penelitian tersebut menunjukkan bahwa algoritme ELM cocok diterapkan pada data yang berbentuk *time series*, dengan nilai MSE yang dihasilkan cukup baik, yakni 0.183479.

Berdasarkan referensi dari beberapa penelitian diatas, maka algoritme ELM akan digunakan dalam penelitian ini karena algoritme tersebut dapat memberikan hasil yang cukup baik dan memiliki waktu pelatihan yang sangat cepat dibandingkan algoritme peramalan lainnya.

Tabel 2.1 Tabel Kajian Pustaka

No.	Judul	Penulis	Kajian Pustaka
1	Extreme Learning Machine for Regression and Multiclass Classification	Huang, et al., 2012	Algoritme ELM untuk regresi dan klasifikasi multiklas
2	Wind Speed Prediction with Extreme Learning Machine	Elizabeta Lazarevska, 2016	Algoritme ELM untuk prediksi kecepatan angin
3	A Comparison of Extreme Learning Machine and Backpropagation Trained Feed-Forward Networks Processing the MNIST Database	Chazal, et al., 2015	Perbandingan Algoritme ELM dan Backpropagation
4	Prediksi Penjualan Mi Menggunakan Metode <i>Extreme Learning Machine</i> (ELM) di Kober Mie Setan Cabang Soekarno Hatta	Giusti, et al., 2017	Penerapan algoritme ELM untuk prediksi penjualan Mi

2.2 Saham

Saham adalah tanda bukti kepemilikan seseorang terhadap sebuah perusahaan (Barus, Christina, 2014). Semakin banyak saham yang dimiliki seseorang dalam sebuah perusahaan, maka semakin besar juga persentase kepemilikan orang tersebut di perusahaan yang bersangkutan. Saham diperjual belikan dalam sebuah pasar saham, di mana masyarakat dapat menjual ataupun membeli saham perusahaan apapun yang tercatat di Bursa Efek Indonesia.

Dalam setiap indeks saham, terdapat parameter yang dapat digunakan untuk meramal harga saham kedepannya. Beberapa parameter tersebut adalah *Open*, *High*, *Low* dan *Price*. *Open* menunjukkan harga saham pada saat bursa saham dibuka dalam satu hari, *High* menunjukkan harga saham tertinggi pada hari tersebut, *Low* menunjukkan harga saham terendah pada hari tersebut dan *Price* menunjukkan harga saham pada saat perdagangan ditutup.

Contoh data saham harian dapat dilihat pada Tabel 2.2. Data yang digunakan sebagai contoh adalah data saham harian Bank Mandiri.

Tabel 2.2 Data saham harian

Date	Price	Open	High	Low
29-Dec-17	8,000	8,050	8,075	7,800
28-Dec-17	7,950	7,975	8,100	7,850
27-Dec-17	8,000	7,675	8,050	7,675
22-Dec-17	7,675	7,625	7,700	7,550

2.3 Peramalan

Peramalan adalah sebuah proses untuk memprediksi data masa depan berdasarkan data masa lalu (Edgar, et al., 2017). Proses peramalan biasanya identik dengan penggunaan data berbentuk *time series*. Salah satu algoritme peramalan yang banyak digunakan saat ini adalah *Extreme Learning Machine* (ELM).

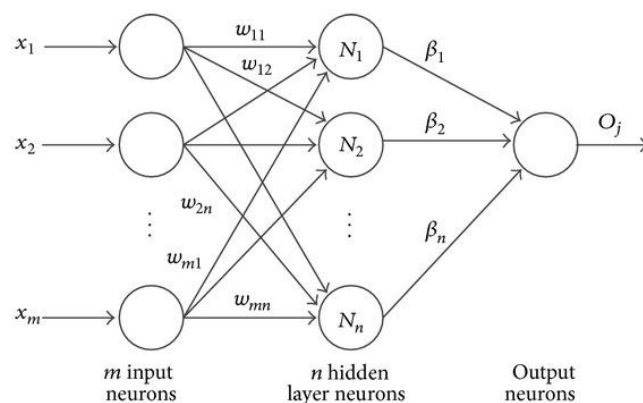
2.3.1 Data Time Series

Data *time series* adalah sekumpulan data sekuensial yang biasanya diukur dalam satuan waktu (Prakaulya, et al., 2017). Salah satu contoh data *time series* adalah data yang akan digunakan dalam penelitian ini, yaitu data harga saham harian Bank Mandiri.

Data berbentuk *time series* biasanya kerap dibandingkan dengan data berbentuk *cross-sectional*. Perbedaan data *time series* dengan data *cross-sectional* adalah data *time series* merupakan kumpulan observasi dari sebuah entitas dalam interval waktu yang berbeda, sedangkan data *cross-sectional* merupakan kumpulan observasi dari beberapa entitas dalam interval waktu yang sama.

2.4 Jaringan Saraf Tiruan

Jaringan Saraf Tiruan adalah sebuah teknologi perangkat lunak yang dimodelkan berdasarkan cara kerja *neuron* pada otak manusia dan memiliki struktur berupa *input layer*, *hidden layer* dan *output layer* (Tee, et al., 2017). Secara umum, arsitektur jaringan saraf tiruan dapat dilihat pada Gambar 2.1. Dalam prosesnya, *input layer* berfungsi sebagai lapisan yang menerima data masukan berupa fitur. *Hidden layer* terdiri dari beberapa neuron yang berfungsi untuk mengekstrak pola yang berhubungan dengan proses yang berjalan. *Output layer* berfungsi untuk merepresentasikan hasil keluaran dari proses yang sudah dilakukan di lapisan lapisan sebelumnya.



Gambar 2.1 Arsitektur jaringan saraf tiruan

Sumber: (You, et al., 2014)

Dalam jaringan saraf tiruan juga terdapat beberapa istilah yang sering digunakan, diantaranya adalah bias dan fungsi aktivasi. Bias digunakan agar bobot yang dipelajari oleh jaringan saraf tiruan menjadi lebih bervariasi yang berarti dapat meningkatkan performa dari jaringan itu sendiri. Fungsi aktivasi berfungsi untuk mengkalkulasikan bobot dari nilai *input* dan akan memetakan bobot bobot tersebut dalam *range* 0 - 1, -1 - 1 atau nilai yang tak terhingga tergantung dari fungsi aktivasi yang digunakan.

2.4.1 Algoritme *Extreme Learning Machine*

Extreme Learning Machine adalah salah satu jenis algoritme pembelajaran dalam sebuah jaringan saraf tiruan. Algoritme ini tidak memiliki *feedback error* layaknya Backpropagation. Algoritme ini termasuk dalam *single-layer feedforward neural network (SLFNs)* dimana nilai bobot dan bias dibangkitkan secara acak (Huang, et al., 2012). Kelebihan algoritme ini dibandingkan algoritme pembelajaran lainnya terletak pada waktu pembelajaran yang lebih cepat.

2.4.2 Tahap Pelatihan *Extreme Learning Machine*

1. Langkah pertama yang dilakukan adalah melakukan normalisasi data agar data yang dihasilkan berada dalam rentang nilai (0,1) atau (-1,1). Hal ini dilakukan agar pada saat tahap aktivasi, data yang digunakan berada dalam range yang wajar. Normalisasi data dapat dilakukan menggunakan rumus berikut:

$$Z = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (2.1)$$

Keterangan:

Z = Nilai hasil normalisasi

Xi = Nilai data index ke-I pada setiap fitur

Min(x) = Nilai terkecil dalam satu fitur

Max(x) = Nilai terbesar dalam satu fitur

2. Membangkitkan bobot input matriks W dimana W adalah hubungan antara neuron ke- i dari input layer dan neuron ke- j dari hidden layer (Li, 2016). Nilai yang dibangkitkan berada dalam rentang nilai $(-1, 1)$.

Kemudian matriks inisialisasi dihitung dengan rumus berikut:

$$H_{init} = (X * W^T) + b \quad (2.2)$$

Keterangan:

X = Matriks input layer

W^T = Matriks transpose W

H_{init} = Matriks inisialisasi

b = Nilai bias

3. Memasukkan fungsi aktivasi terhadap matriks inisialisasi dengan rumus berikut:

$$H = 1/(1 + \exp(-H_{init})) \quad (2.3)$$

Keterangan:

H = Matriks hasil fungsi aktivasi

4. Menghitung nilai H^+ yang didapatkan melalui rumus berikut:

$$H^+ = (H^T * H)^{-1} * H^T \quad (2.4)$$

5. Kemudian menghitung nilai β dengan rumus:

$$\beta = H^+ * Y \quad (2.5)$$

Keterangan:

β = Vektor nilai Beta estimasi

H^+ = Nilai *Moore-Penrose generalized inverse* dari matriks H

Y = Vektor output layer

2.4.3 Tahap Pengujian *Extreme Machine Learning*

1. Setelah didapatkan nilai Beta dari tahap pelatihan, kemudian menghitung nilai Y hasil peramalan dengan rumus:

$$Y_{pred} = \beta * H \quad (2.6)$$

Keterangan:

Y_{pred} = Vektor hasil peramalan

β = Vektor nilai Beta estimasi

H = Matriks hasil fungsi aktivasi

2. Hasil peramalan kemudian didenormalisasi untuk mendapatkan nilai real dari peralaman tersebut dengan menggunakan rumus:

$$Denormalisasi = x(max - min) + min \tag{2.7}$$

Keterangan:

X = Data yang akan didenormalisasi

Max = Nilai maksimum fitur yang bersangkutan

Min = Nilai minimum fitur yang bersangkutan

2.5 Evaluasi

Setelah pengujian dilakukan, maka hasil yang didapatkan harus dievaluasi untuk mengetahui seberapa besar tingkat kesalahan yang dihasilkan algoritme ini. Salah satu teknik untuk evaluasi tingkat kesalahan adalah menghitung nilai *Mean Absolute Percentage Error* (MAPE). MAPE adalah salah satu cara pengukuran nilai akurasi dalam persentase error (Tee, et al., 2017). Evaluasi tersebut dilakukan dengan rumus berikut:

$$MAPE = \frac{1}{N} \sum_{i=1}^n \frac{|Y - Y_{pred}|}{Y} \tag{2.8}$$

Keterangan:

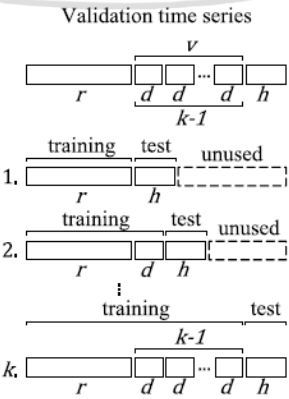
N = Banyaknya data

Y = Nilai Y asli

Ypred = Nilai Y prediksi

2.6 Cross Validation

Cross Validation adalah sebuah metode yang dilakukan untuk menguji kemampuan prediksi sebuah model (Fonseca-Delgado, Gomez-Gil, 2013). Pada salah satu metode *cross validation*, *k-Fold*, data yang digunakan dibagi menjadi beberapa bagian (*fold*) yang berukuran kurang lebih sama.



Gambar 2.2 Proses *k-Fold* pada model peramalan *time-series*

Sumber: (Fonseca-Delgado, Gomez-Gil, 2013)

Gambar 2.2 menunjukkan proses *k-Fold cross validation* pada model peramalan dengan data *time-series*. Data yang digunakan untuk proses pelatihan dilabeli dengan huruf *r*, sedangkan data yang dilabeli dengan huruf *h* adalah data yang digunakan untuk proses pengujian.



BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan mengenai alur metodologi penelitian yang dilakukan pada penelitian ini.

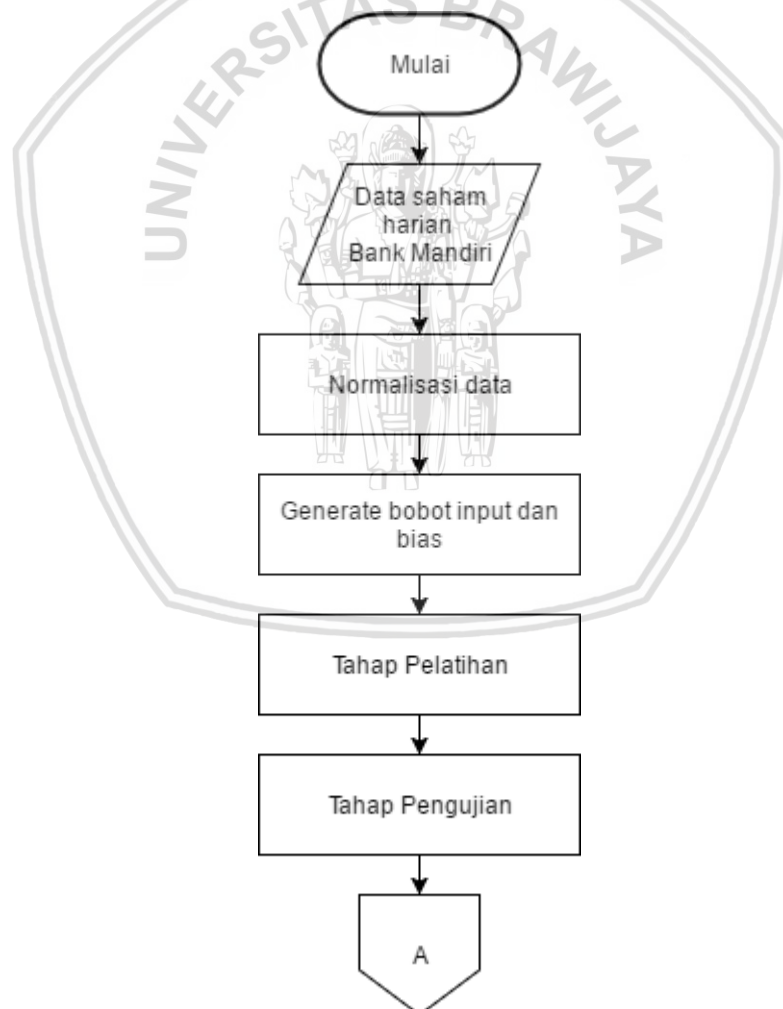
3.1 Tipe Penelitian

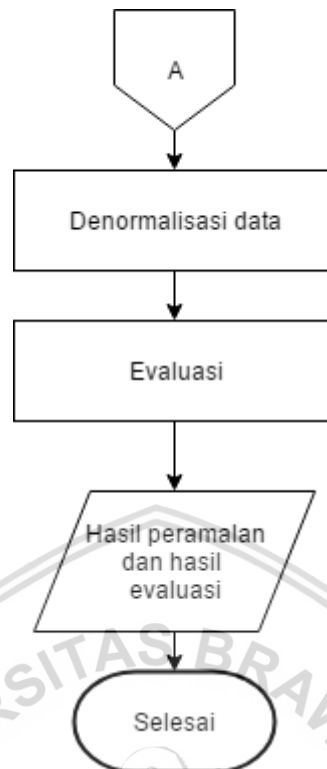
Tipe penelitian yang digunakan pada penelitian ini adalah non-implementatif analitik, yang mana kegiatan yang dilakukan adalah menganalisis derajat hubungan antar elemen dalam objek yang diteliti.

3.2 Strategi dan Rancangan Penelitian

Strategi dan rancangan dari penelitian yang dilakukan akan dijelaskan dalam sub-bab berikut.

3.2.1 Metode Secara Umum





Gambar 3.1 Diagram alir gambaran umum sistem

Dalam penelitian ini, akan dibuat sebuah program yang mengimplementasikan metode *Extreme Learning Machine* (ELM) untuk peramalan saham. Gambar 3.1 menunjukkan diagram alir gambaran umum sistem. Data masukkan yang digunakan adalah data saham harian Bank Mandiri. Data tersebut kemudian dinormalisasi menggunakan *Min-Max Normalization*. Kemudian sistem akan membangkitkan bobot input dan bobot bias secara random. Data yang sudah dinormalisasi beserta bobot yang sudah dibangkitkan kemudian akan memasuki tahap pelatihan. Dari tahap pelatihan akan didapatkan nilai β yang akan digunakan dalam tahap pengujian. Data yang dihasilkan dari tahap pengujian kemudian akan didenormalisasi untuk kemudian dievaluasi menggunakan *Mean Absolute Percentage Error* (MAPE). Keluaran dari sistem ini adalah hasil peramalan yang sudah didenormalisasi serta nilai *error*-nya.

3.2.2 Objek Penelitian

Objek yang disertakan dalam penelitian ini adalah Bank Mandiri. Data saham harian Bank Mandiri pada rentang waktu 16 Maret 2012 – 29 Desember 2017 akan diambil dengan parameter yang digunakan untuk peramalan adalah *Open*, *High* dan *Low*. Parameter *Open* menunjukkan harga saham pada waktu perdagangan dibuka, *High* menunjukkan harga saham tertinggi pada satu hari perdagangan, sedangkan *Low* menunjukkan harga saham tertendah pada satu hari perdagangan. Ketiga parameter ini digunakan untuk meramal *Price*, yaitu harga saham pada saat perdagangan hari berikutnya ditutup.

3.2.3 Lokasi Penelitian

Penelitian ini dilakukan di laboratorium riset komputasi cerdas Fakultas Ilmu Komputer Universitas Brawijaya.

3.2.4 Metode Pengumpulan Data

Data saham harian Bank Mandiri yang digunakan diunduh melalui alamat website <https://www.investing.com/equities/bank-mandiri-t>. Data yang diunduh memiliki format *Comma Separated Value* (CSV). Data tersebut merupakan data sekunder karena tidak diperoleh langsung dari pihak Bank Mandiri melainkan dari pihak ketiga.

3.2.5 Peralatan Pendukung

Penelitian ini dilakukan pada komputer dengan spesifikasi sebagai berikut:

- Processor : Intel® Core™ i7-770 CPU @ 3.60GHz
- RAM : 8 Gigabyte
- OS : Windows 10 Pro 64-bit

Selain itu, Bahasa pemrograman Python 2.7 akan digunakan untuk mengimplementasikan metode yang dipilih dalam penelitian ini.

3.3 Jadwal Kegiatan

Penelitian ini dijadwalkan untuk dapat selesai dalam kurun waktu empat bulan. Detail dari rencana jadwal kegiatan penelitian ini dapat dilihat pada Tabel 3.1.

Tabel 3.1 Rencana Jadwal Kegiatan

No.	Kegiatan	Bulan			
		1	2	3	4
1	Studi Pustaka				
2	Pengumpulan Data				
3	Perancangan				
4	Implementasi				
5	Pengujian dan Analisis				
6	Kesimpulan dan Saran				

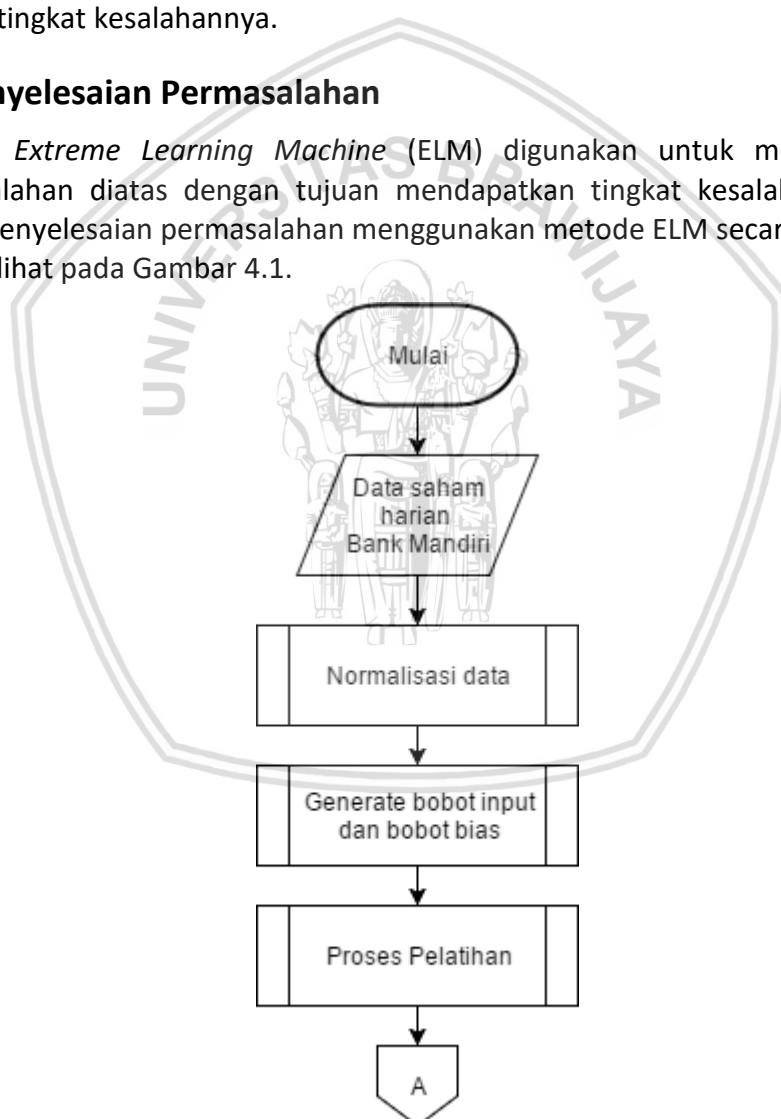
BAB 4 PERANCANGAN DAN IMPLEMENTASI

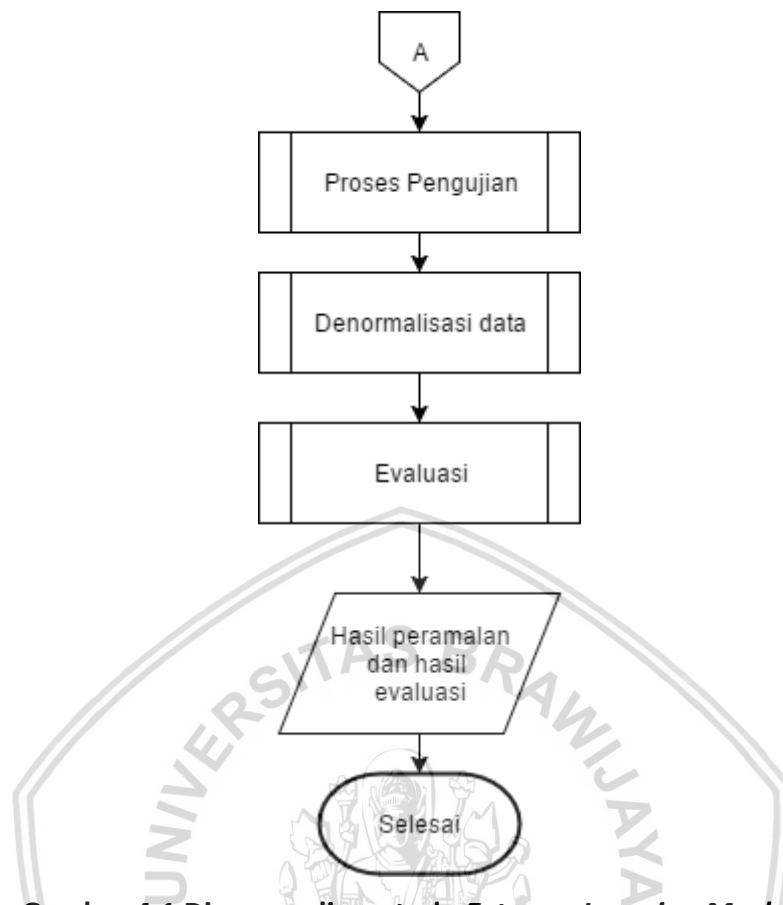
4.1 Formulasi Permasalahan

Permasalahan yang akan diselesaikan pada penelitian ini adalah peramalan saham harian Bank Mandiri dengan menggunakan metode *Extreme Learning Machine* (ELM). Data yang digunakan untuk peramalan adalah data saham harian Bank Mandiri dari tanggal 16 Maret 2012 – 29 Desember 2017, sedangkan parameter-parameter yang digunakan diantaranya adalah *Open*, *High*, *Low*, jumlah neuron dan fungsi aktivasi. Perbandingan data latih dan data uji yang digunakan pada penelitian ini adalah 70% data latih dan 30% data uji. Hasil peramalan kemudian akan dievaluasi menggunakan *Mean Absolute Percentage Error* (MAPE) untuk melihat tingkat kesalahannya.

4.2 Penyelesaian Permasalahan

Metode *Extreme Learning Machine* (ELM) digunakan untuk menyelesaikan permasalahan diatas dengan tujuan mendapatkan tingkat kesalahan terkecil. Proses penyelesaian permasalahan menggunakan metode ELM secara garis besar dapat dilihat pada Gambar 4.1.





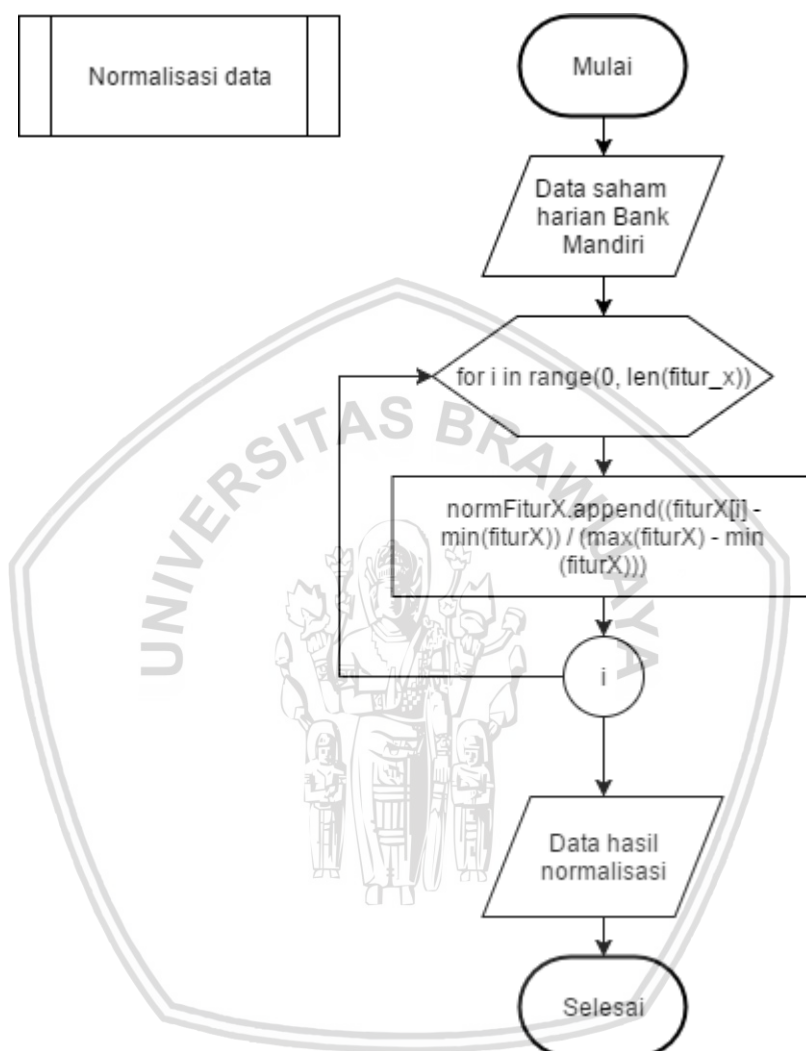
Gambar 4.1 Diagram alir metode *Extreme Learning Machine*

Berdasarkan Gambar 4.1, langkah-langkah metode ELM adalah sebagai berikut:

1. Algoritme mendapatkan masukan berupa data saham harian Bank Mandiri.
2. Data tersebut kemudian dinormalisasi menggunakan Persamaan 2.1.
3. Melakukan pembangkitan nilai bobot input dan bias yang akan digunakan dalam tahap pelatihan dan pengujian.
4. Data latih yang sudah dinormalisasi kemudian masuk dalam tahap pelatihan. *Output* dari tahap pelatihan adalah nilai β .
5. Data uji yang sudah dinormalisasi masuk dalam tahap pengujian. *Output* dari tahap pelatihan adalah hasil peramalan yang belum didenormalisasi.
6. Hasil peramalan kemudian akan didenormalisasi untuk mendapatkan nilai aslinya.
7. Nilai asli dari hasil peramalan kemudian akan dievaluasi menggunakan *Mean Absolute Percentage Error* untuk mengetahui tingkat kesalahan dari algoritme ini.
8. Hasil akhir yang didapat adalah nilai asli dari hasil peramalan serta tingkat kesalahan algoritme ini.

4.2.1 Normalisasi Data

Normalisasi data dilakukan karena data yang digunakan pada penelitian ini berada pada *range* yang besar. Normalisasi data pada penelitian ini menggunakan metode *Min-Max Normalization* yang dapat dilihat pada Persamaan 2.1. Gambar 4.2 menunjukkan langkah-langkah normalisasi data yang dilakukan.

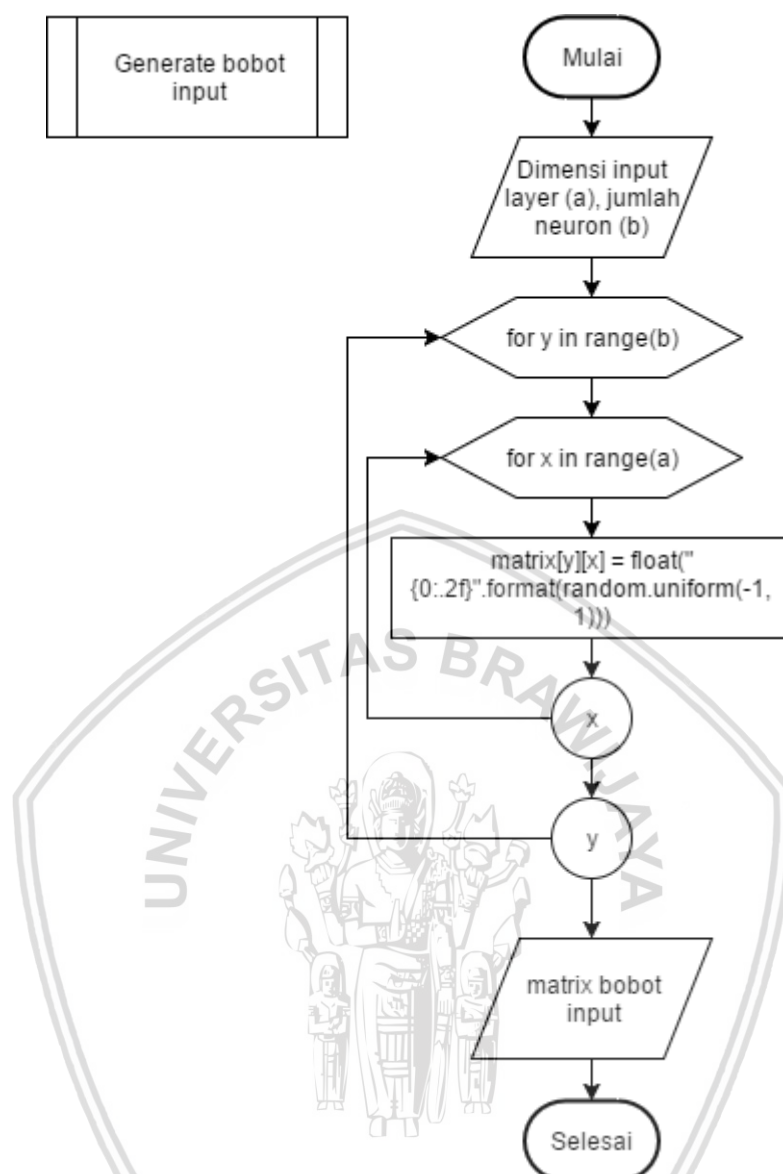


Gambar 4.2 Diagram alir normalisasi data

Data saham harian Bank Mandiri yang terdiri dari 4 fitur yaitu *Price*, *Open*, *High* dan *Low* menjadi masukan pada proses ini. Langkah pertama yang harus dilakukan adalah mencari nilai maksimum dan minimum dari masing-masing fitur. Setelah nilai maksimum dan minimum masing-masing fitur sudah ditemukan, setiap data dari masing-masing fitur dinormalisasi menggunakan Persamaan 2.1. Hasil dari proses ini adalah data yang sudah dinormalisasi.

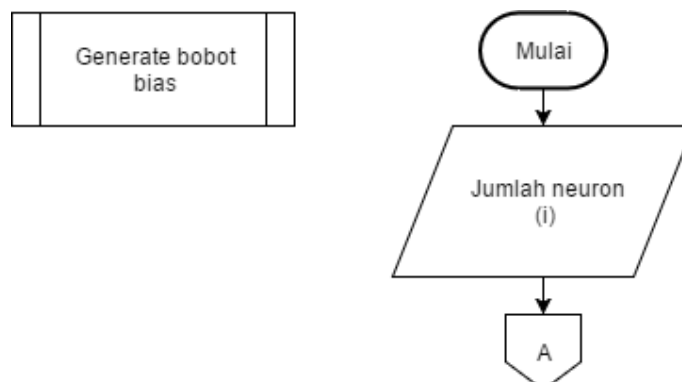
4.2.2 Pembangkitan Bobot Input dan Bobot Bias

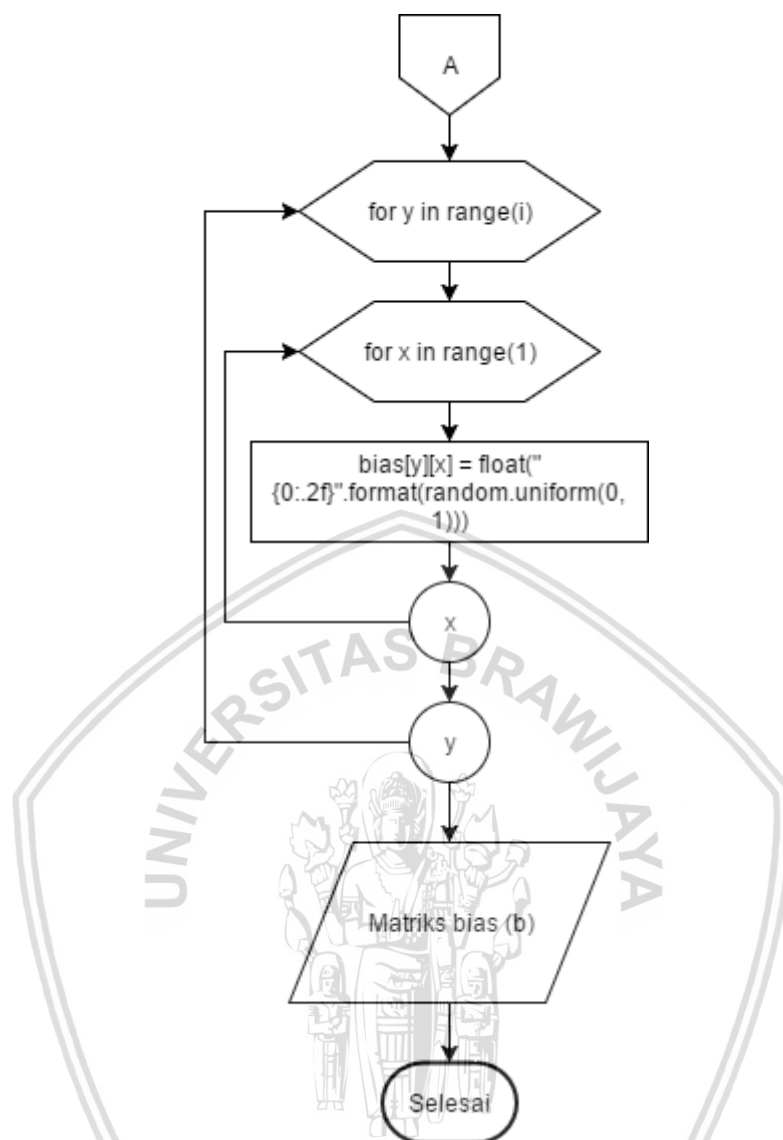
Setelah melakukan normalisasi, langkah berikutnya yang harus dilakukan adalah membangkitkan matriks nilai bobot input dan bobot bias. Gambar 4.3 menunjukkan langkah-langkah untuk membangkitkan matriks nilai bobot input.



Gambar 4.3 Diagram alir pembangkitan matriks bobot input

Matriks bobot input ini memiliki dimensi sebesar jumlah neuron x dimensi input layer. Nilai yang dibangkitkan untuk matriks bobot ini berada dalam *range* -1 hingga 1. Langkah selanjutnya setelah membangkitkan matriks bobot input adalah membangkitkan matriks bobot bias.



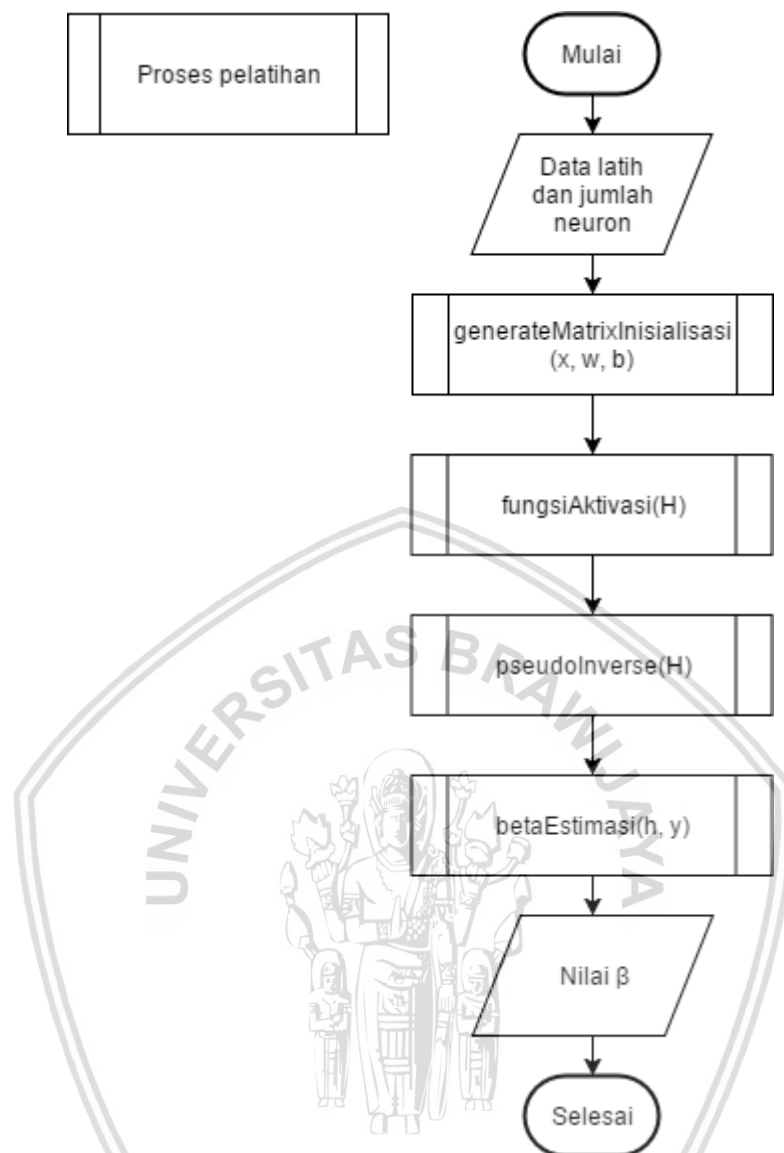


Gambar 4.4 Diagram alir pembangkitan matriks bobot bias

Gambar 4.4 menunjukkan langkah-langkah untuk membangkitkan nilai matriks bobot bias. Matriks ini memiliki dimensi sebesar 1 x jumlah neuron. Sama seperti matriks bobot input, matriks bobot bias memiliki nilai antara 0 hingga 1.

4.2.3 Tahap Pelatihan

Setelah data saham harian Bank Mandiri dinormalisasi dan matriks bobot input serta bias sudah dibangkitkan, maka data yang digunakan akan dibagi menjadi dua, yaitu data latih dan data uji. Data latih kemudian akan menjadi masukan dalam tahap pelatihan untuk mendapatkan nilai β yang nantinya akan digunakan pada tahap pengujian. Langkah-langkah proses pelatihan dapat dilihat pada Gambar 4.5.



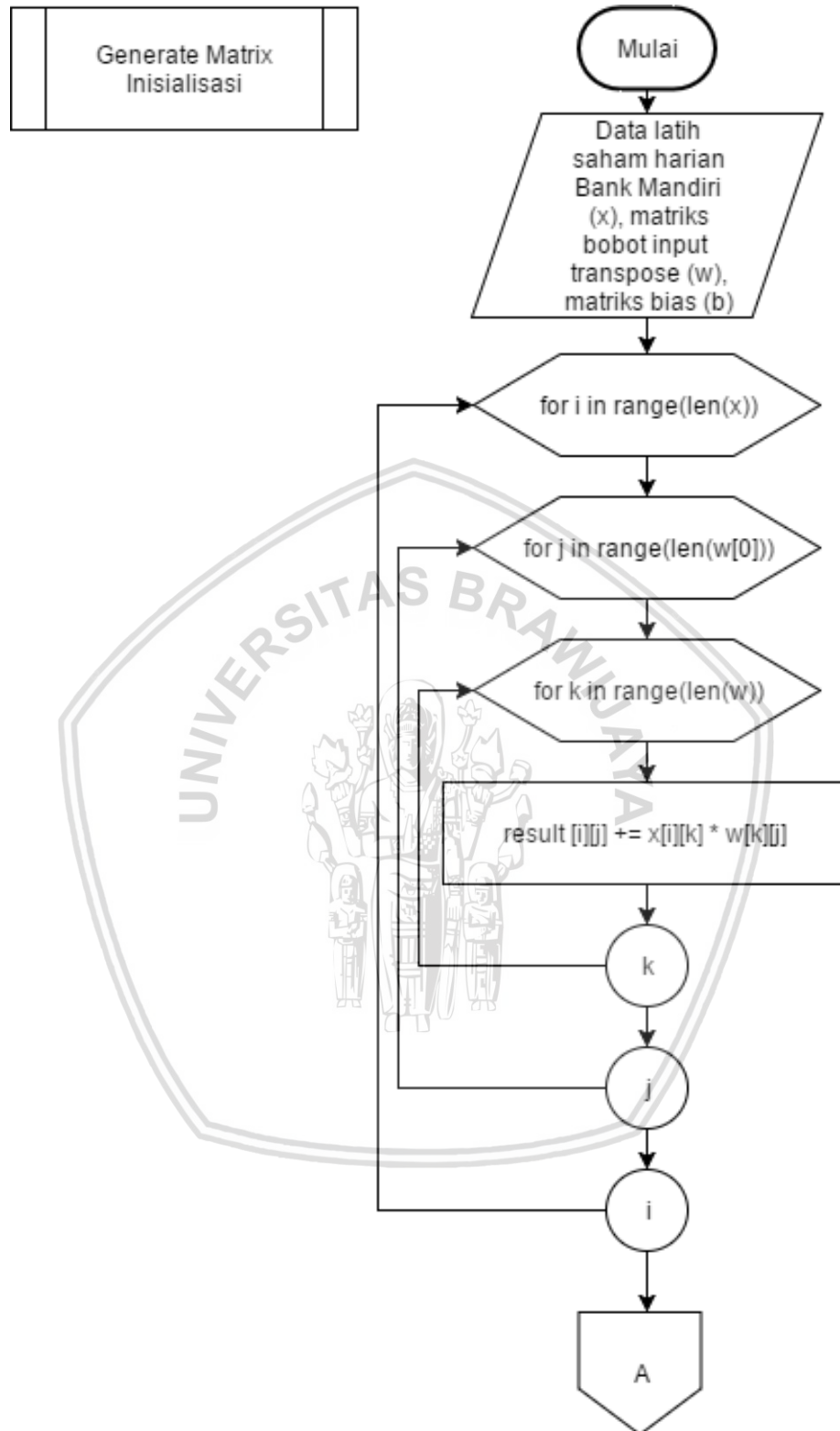
Gambar 4.5 Diagram alir proses pelatihan

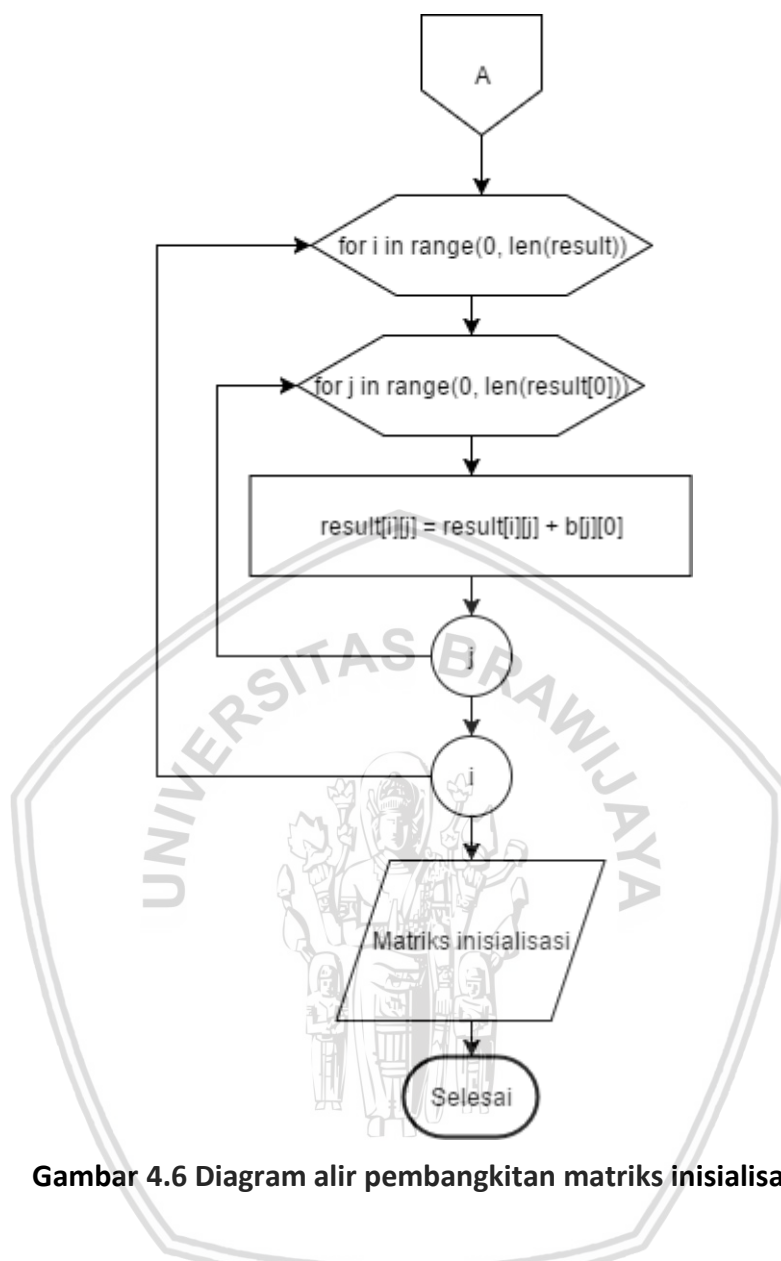
Berikut adalah langkah-langkah proses pelatihan secara detail:

1. Membangkitkan matriks inisialisasi menggunakan masukan yang berupa data latih saham harian Bank Mandiri, matriks bobot input dan matriks bobot bias. Matriks ini dibangkitkan menggunakan Persamaan 2.2.
2. Matriks inisialisasi tersebut kemudian diaktivasi menggunakan fungsi aktivasi yang telah ditentukan (sigmoid, sin, tan, dll).
3. Melakukan *pseudo-inverse* terhadap matriks yang sudah diaktivasi menggunakan Persamaan 2.4.
4. Menghitung nilai β menggunakan Persamaan 2.5.

4.2.3.1 Membangkitkan Matriks Inisialisasi

Langkah-langkah untuk membangkitkan Matriks Inisialisasi dapat dilihat pada Gambar 4.6.





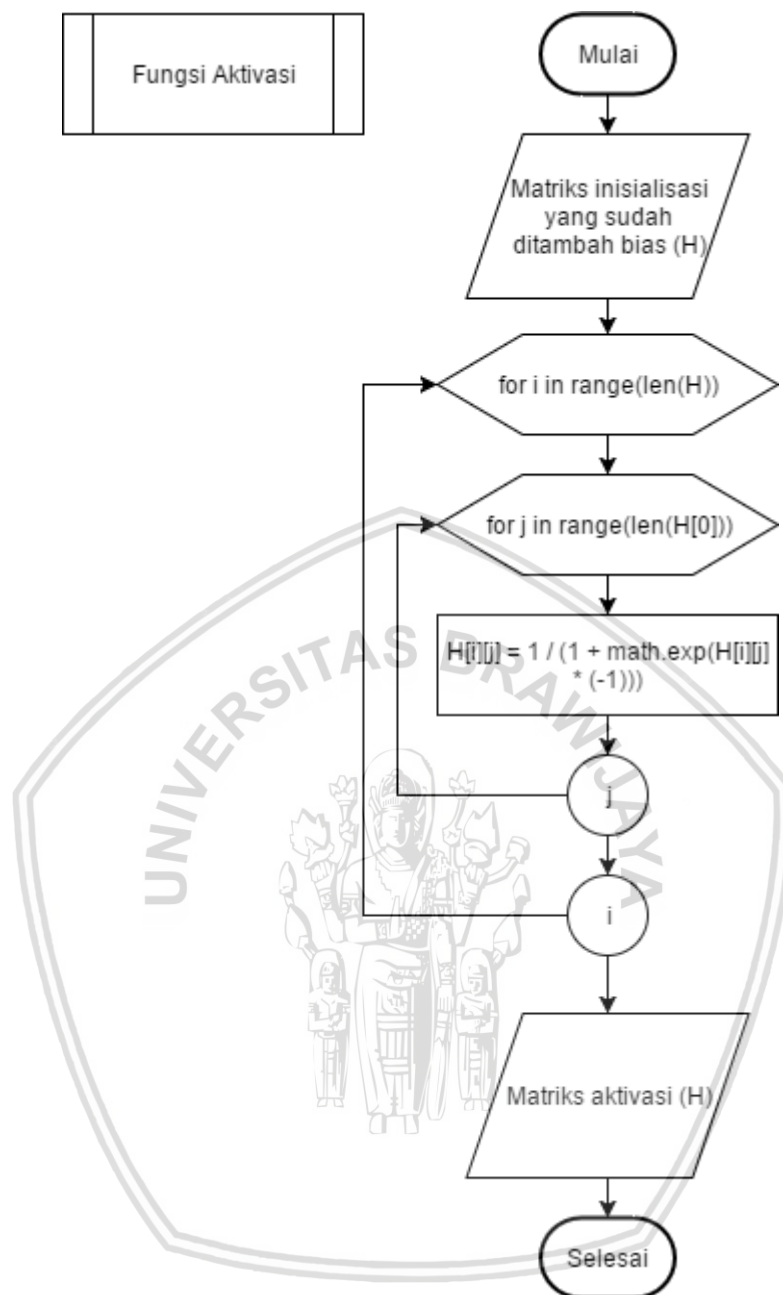
Gambar 4.6 Diagram alir pembangkitan matriks inisialisasi

Berikut adalah langkah-langkah yang dilakukan untuk membangkitkan matriks inisialisasi secara detail:

1. Algoritme menerima masukan berupa data latih saham harian Bank Mandiri yang sudah dinormalisasi, matriks bobot input dan matriks bobot bias.
2. Matriks inisialisasi dihitung menggunakan Persamaan 2.2
3. Keluaran berupa matriks inisialisasi nantinya akan digunakan sebagai masukan untuk proses fungsi aktivasi.

4.2.3.2 Fungsi Aktivasi

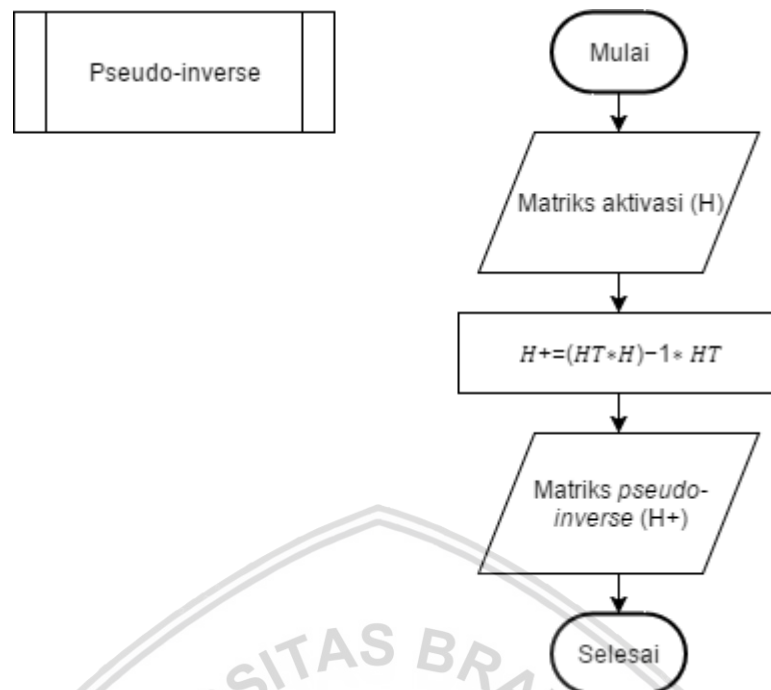
Matriks inisialisasi yang didapatkan kemudian diaktivasi menggunakan sebuah fungsi aktivasi. Gambar 4.7 menunjukkan contoh aktivasi matriks inisialisasi menggunakan fungsi aktivasi sigmoid.



Gambar 4.7 Diagram alir fungsi aktivasi

4.2.3.3 Pseudo-Inverse

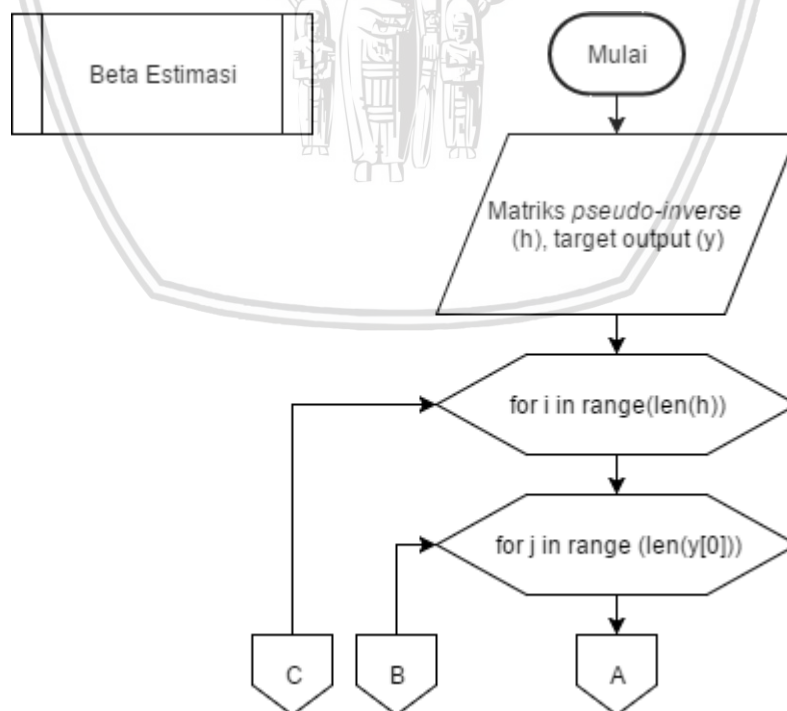
Matriks yang sudah diaktivasi kemudian memasuki tahap *pseudo-inverse* untuk menghasilkan bobot optimal yang nantinya akan digunakan untuk menghitung nilai β . Langkah-langkah untuk melakukan *pseudo-inverse* dapat dilihat pada Gambar 4.8.

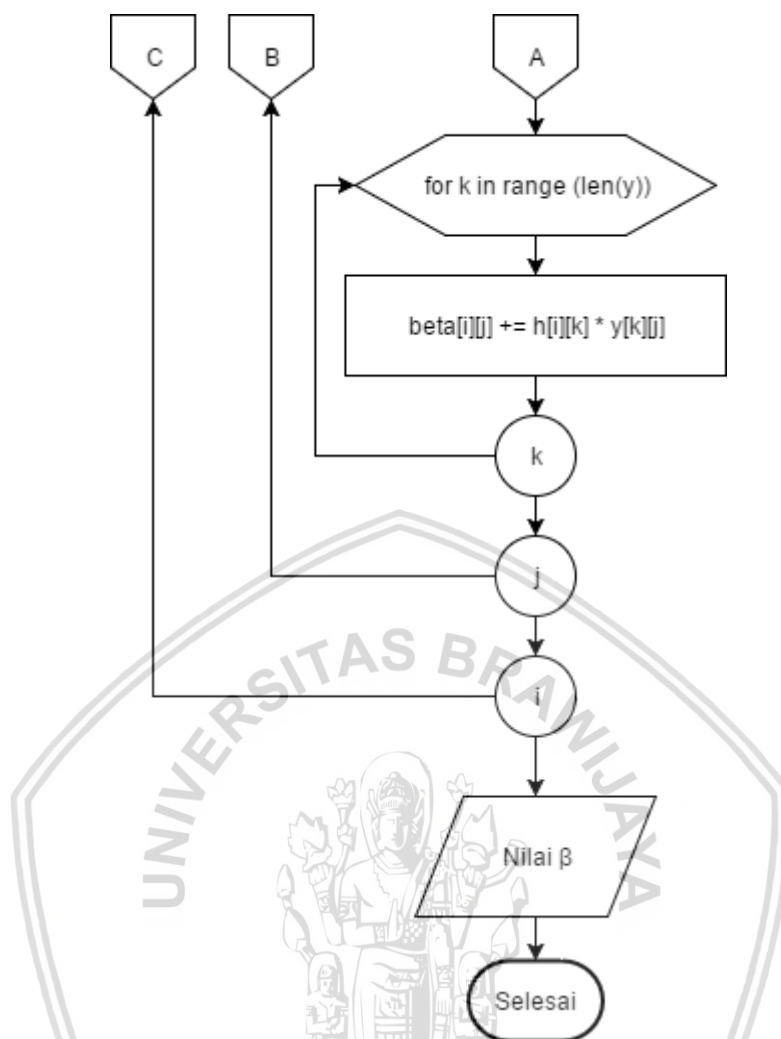


Gambar 4.8 Diagram alir *pseudo-inverse*

4.2.3.4 Menghitung nilai β

Matriks *pseudo-inverse* kemudian dikalikan dengan vektor target untuk mendapatkan nilai β yang nantinya akan digunakan dalam tahap pengujian. Langkah-langkah untuk mendapatkan nilai β dapat dilihat pada Gambar 4.9.





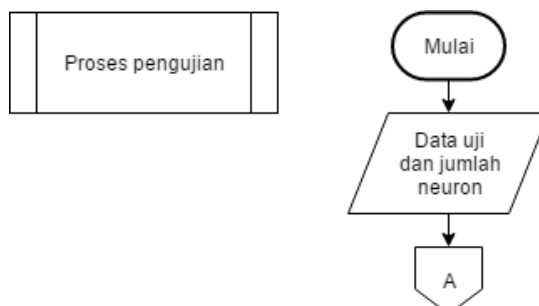
Gambar 4.9 Diagram alir menghitung nilai β

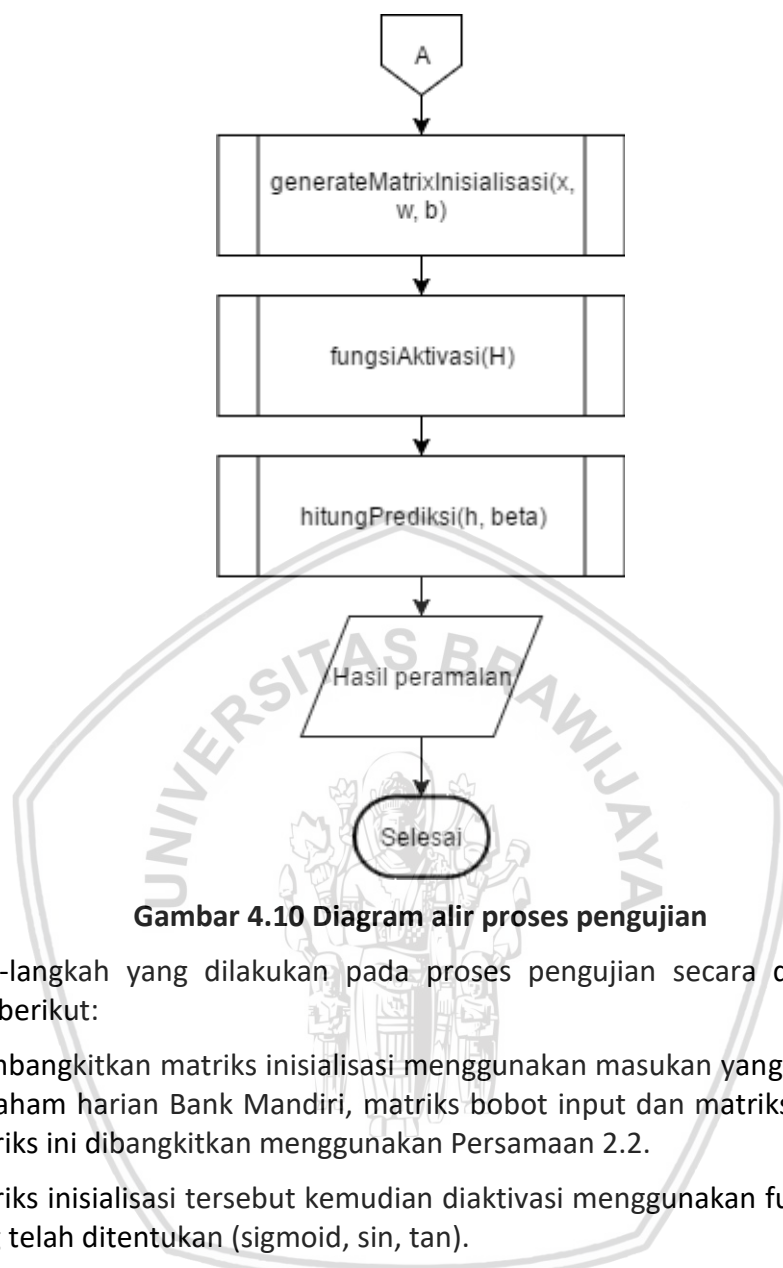
Setelah nilai β didapatkan, maka tahap pelatihan selesai dan sekarang algoritme akan memasuki tahap pengujian.

4.2.4 Tahap Pengujian

Tahap pengujian dilakukan untuk mendapatkan hasil peramalan dengan masukan berupa data uji saham harian Bank Mandiri, matriks bobot input, matriks bobot bias serta nilai β yang didapatkan dari tahap pelatihan.

Secara garis besar, langkah-langkah yang dilakukan pada tahap pengujian dapat dilihat pada Gambar 4.10.





Gambar 4.10 Diagram alir proses pengujian

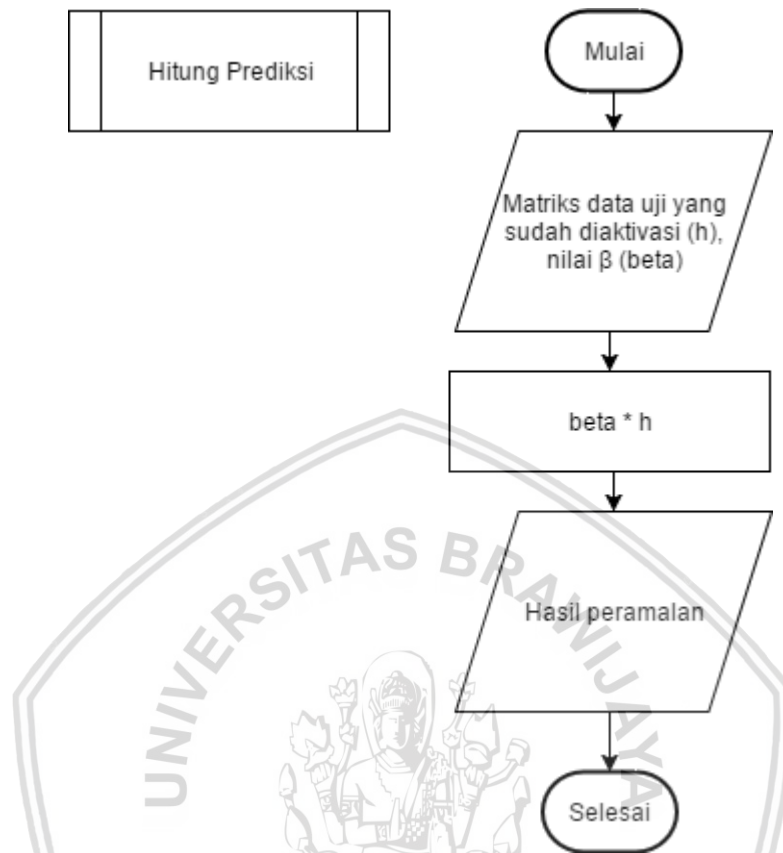
Langkah-langkah yang dilakukan pada proses pengujian secara detail adalah sebagai berikut:

1. Membangkitkan matriks inisialisasi menggunakan masukan yang berupa data uji saham harian Bank Mandiri, matriks bobot input dan matriks bobot bias. Matriks ini dibangkitkan menggunakan Persamaan 2.2.
2. Matriks inisialisasi tersebut kemudian diaktivasi menggunakan fungsi aktivasi yang telah ditentukan (sigmoid, sin, tan).
3. Menghitung hasil peramalan dengan menggunakan masukan berupa matriks inisialisasi yang sudah diaktivasi dan nilai β yang didapatkan pada tahap pelatihan.
4. *Output* dari tahap pengujian adalah hasil peramalan yang belum dinormalisasi.

4.2.4.1 Proses Menghitung Hasil Peramalan

Proses untuk menghitung peramalan menggunakan masukan berupa matriks inisialisasi data uji yang sudah diaktivasi dan nilai β . Hasil peramalan tersebut nantinya akan memasuki tahap dinormalisasi untuk keperluan evaluasi.

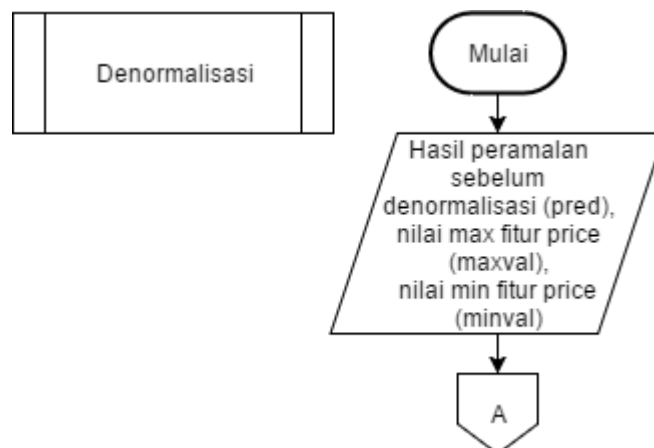
Langkah-langkah untuk menghitung hasil peramalan dapat dilihat pada Gambar 4.11.

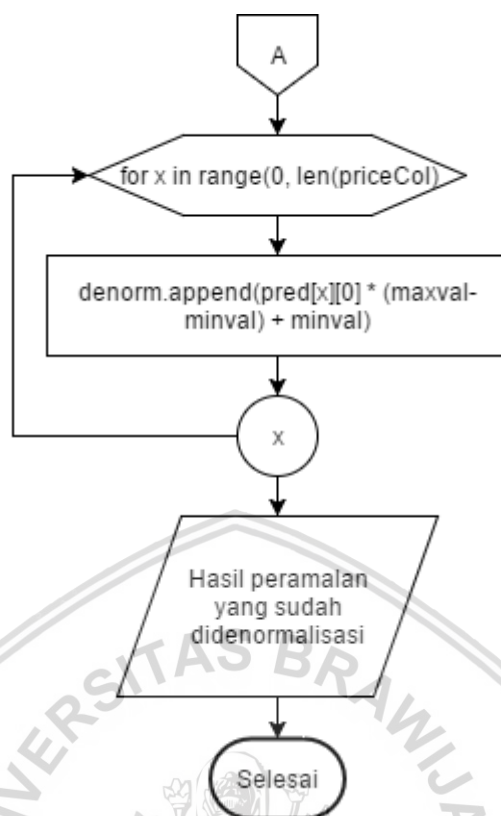


Gambar 4.11 Diagram alir proses menghitung hasil peramalan

4.2.5 Denormalisasi

Hasil peramalan dari tahap pelatihan kemudian akan didenormalisasi untuk mendapatkan nilai asli dari peramalan tersebut. Proses denormalisasi dilakukan terhadap setiap data hasil peramalan dengan menggunakan Persamaan 2.7. Langkah-langkah yang dilakukan untuk denormalisasi data dapat dilihat pada Gambar 4.12.





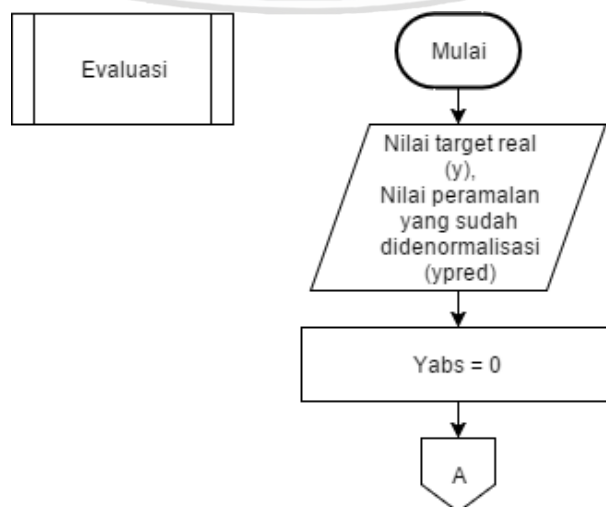
Gambar 4.12 Diagram alir proses denormalisasi

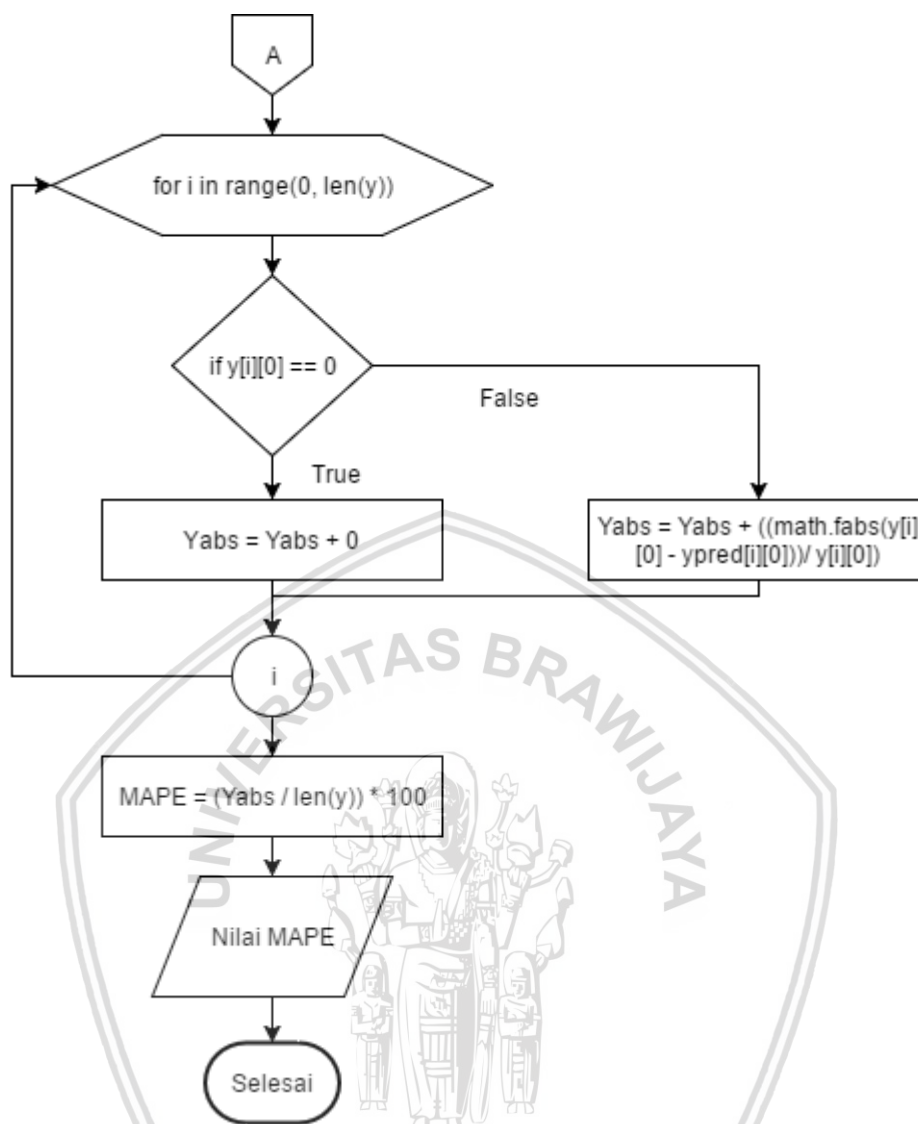
Output dari proses denormalisasi kemudian akan menjadi masukan untuk tahap evaluasi.

4.2.6 Evaluasi

Evaluasi dilakukan untuk melihat tingkat kesalahan dari algoritme yang digunakan. Pada penelitian ini, *Mean Absolute Percentage Error* (MAPE) akan digunakan sebagai metode evaluasi. *Output* dari tahap evaluasi ini adalah nilai tingkat kesalahan dalam bentuk persentase.

Langkah-langkah untuk evaluasi dapat dilihat pada Gambar 4.13.





Gambar 4.13 Diagram alir proses evaluasi

Setelah evaluasi dilakukan, maka keseluruhan proses pada algoritme ini telah selesai dengan *output* akhir berupa hasil peramalan yang sudah didenormalisasi dan nilai tingkat kesalahan dalam bentuk nilai MAPE.

4.3 Perhitungan Manual

Contoh perhitungan manual yang dilakukan dalam penelitian ini akan menggunakan data sampel sebanyak sepuluh buah, jumlah neuron sebanyak 3 buah dan fungsi aktivasi yang digunakan adalah fungsi aktivasi sigmoid. Data yang digunakan sebagai data sampel dapat dilihat pada Tabel 4.1.

Tabel 4.1 Data saham harian Bank Mandiri

Date	Price + 1	Open	High	Low
13-Dec-17	7,475	7,400	7,500	7,400
14-Dec-17	7,425	7,500	7,550	7,425

15-Dec-17	7,500	7,400	7,475	7,300
18-Dec-17	7,475	7,325	7,500	7,300
19-Dec-17	7,450	7,375	7,525	7,350
20-Dec-17	7,625	7,525	7,525	7,375
21-Dec-17	7,675	7,550	7,700	7,500
22-Dec-17	8,000	7,625	7,700	7,550
27-Dec-17	7,950	7,675	8,050	7,675
28-Dec-17	8,000	7,975	8,100	7,850

Keterangan:

Price + 1 = Harga tutup hari esok

Open = Harga buka hari ini

High = Harga tertinggi hari ini

Low = Harga terendah hari ini

Data sampel yang berjumlah sepuluh buah kemudian dibagi dua menjadi data latih dan data uji dengan komposisi tujuh buah data latih dan tiga buah data uji. Data yang digunakan sebagai data latih dan data uji dapat dilihat pada Tabel 4.2 dan Tabel 4.3.

Tabel 4.2 Data latih

Date	Price + 1	Open	High	Low
13-Dec-17	7,475	7,400	7,500	7,400
14-Dec-17	7,425	7,500	7,550	7,425
15-Dec-17	7,500	7,400	7,475	7,300
18-Dec-17	7,475	7,325	7,500	7,300
19-Dec-17	7,450	7,375	7,525	7,350
20-Dec-17	7,625	7,525	7,525	7,375
21-Dec-17	7,675	7,550	7,700	7,500

Tabel 4.3 Data uji

Date	Price + 1	Open	High	Low
22-Dec-17	8,000	7,625	7,700	7,550
27-Dec-17	7,950	7,675	8,050	7,675
28-Dec-17	8,000	7,975	8,100	7,850

Langkah pertama yang dilakukan adalah normalisasi data. Pada penelitian ini, rumus normalisasi yang digunakan adalah *Min-Max Normalization* yang dapat dilihat pada Persamaan 2.1. Tabel 4.4 menunjukkan contoh data maksimum dan minimum pada fitur *Open*.

Tabel 4.4 Nilai maksimum dan minimum fitur *Open*

Max	7,550
Min	7,325

Nilai di atas kemudian akan digunakan untuk melakukan normalisasi pada setiap data fitur *Open*. Berikut adalah contoh perhitungan normalisasi pada data indeks pertama fitur *Open*:

$$X_{1,1} = \frac{X_{1,1} - \min}{\max - \min} = \frac{7400 - 7375}{7550 - 7375} = 0.3333$$

Hasil normalisasi data keseluruhan fitur yang digunakan dapat dilihat pada Tabel 4.5.

Tabel 4.5 Normalisasi

Date	Price + 1	Open	High	Low
13-Dec-17	0.20000	0.33333	0.11111	0.50000
14-Dec-17	0.00000	0.77778	0.33333	0.62500
15-Dec-17	0.30000	0.33333	0.00000	0.00000
18-Dec-17	0.20000	0.00000	0.11111	0.00000
19-Dec-17	0.10000	0.22222	0.22222	0.25000
20-Dec-17	0.80000	0.88889	0.22222	0.37500
21-Dec-17	1.00000	1.00000	1.00000	1.00000
22-Dec-17	2.30000	1.33333	1.00000	1.25000
27-Dec-17	2.10000	1.55556	2.55556	1.87500
28-Dec-17	2.30000	2.88889	2.77778	2.75000

Langkah selanjutnya adalah membangkitkan nilai matriks bobot dan matriks bobot bias. Nilai yang dibangkitkan pada kedua matriks tersebut berada pada *range* 1 dan -1. Nilai matriks bobot awal dapat dilihat pada Tabel 4.6.

Tabel 4.6 Matriks bobot

w	1	2	3
1	-0.3	0.39	-0.46
2	-0.41	0.61	-0.58
3	0.15	0.91	-0.94

Matriks bobot awal kemudian ditranspose menjadi matriks baru yang nantinya akan digunakan untuk proses pelatihan. Matriks transpose tersebut dapat dilihat pada Tabel 4.7

Tabel 4.7 Matriks transpose bobot

w	1	2	3
1	-0.3	-0.41	0.15
2	0.39	0.61	0.91
3	-0.46	-0.58	-0.94

Tabel 4.8 menunjukkan matriks bobot bias. Pada contoh perhitungan manual ini, jumlah neuron yang digunakan adalah tiga buah neuron, oleh karena itu bobot bias yang dibangkitkan juga berjumlah tiga buah.

Tabel 4.8 Matriks bobot bias

1	2	3
-0.84	-0.72	0.53

4.3.1 Perhitungan Manual Proses Pelatihan

Langkah pertama yang dilakukan pada proses pelatihan adalah menghitung matriks inisialisasi dengan menggunakan rumus pada Persamaan 2.2. Berikut adalah contoh perhitungan matriks inisialisasi pada indeks 1,1:

$$H_{init\ i,j} = (X * W^T) + b_j$$

$$H_{1,1} = ((0.3333 * -0.3) + (0.1111 * 0.39) + (0.5 * -0.46) + -0.84) \\ = -1.12667$$

Tabel 4.9 menunjukkan nilai matriks inisialisasi.

Tabel 4.9 Matriks inisialisasi

Hinit	1	2	3
1	-1.126666667	-1.078888889	0.211111111
2	-1.230833333	-1.198055556	0.3625
3	-0.94	-0.856666667	0.58
4	-0.796666667	-0.652222222	0.631111111
5	-0.935	-0.820555556	0.530555556
6	-1.1925	-1.166388889	0.513055556
7	-1.21	-1.1	0.65

Langkah berikutnya adalah memasukkan fungsi aktivasi sigmoid terhadap matriks inisialisasi. Berikut adalah contoh perhitungan fungsi aktivasi sigmoid pada matriks inisialisasi indeks 1,1:

$$H(x)_{1,1} = \frac{1}{1 + e^{-x}} = \frac{1}{1 + e^{-(-1.126666667)}} = 0.244777$$

Tabel 4.10 menunjukkan nilai matriks inisialisasi yang sudah diaktivasi menggunakan fungsi aktivasi sigmoid.

Tabel 4.10 Matriks fungsi aktivasi

Aktivasi H(x)	1	2	3
1	0.24477678	0.253716342	0.552582631
2	0.226035606	0.231821303	0.589645479
3	0.280900343	0.298036245	0.641067406

4	0.310739002	0.342488939	0.652741361
5	0.281911425	0.305645744	0.629612677
6	0.232812111	0.23750833	0.625522494
7	0.229701051	0.249739894	0.657010463

Matriks hasil fungsi aktivasi di atas akan digunakan untuk menghitung nilai *Beta* estimasi yang didapatkan menggunakan rumus pada Persamaan 2.5. Untuk dapat menghitung nilai *Beta* estimasi, harus dilakukan perhitungan untuk mendapatkan matriks *Moore-Penrose Generalized Inverse* yang didapatkan menggunakan rumus pada Persamaan 2.4. Untuk itu, matriks hasil fungsi aktivasi—yang selanjutnya akan dilambangkan dengan *H*—perlu di-*transpose* terlebih dahulu. Tabel 4.11 menunjukkan matriks *H transpose*.

Tabel 4.11 Matriks transpose fungsi aktivasi

H^T	1	2	3	4	5	6	7
1	0.244776	0.226035	0.280900	0.310739	0.281911	0.232812	0.229701
2	0.253716	0.231821	0.298036	0.342488	0.305645	0.23750	0.249739
3	0.552582	0.589645	0.641067	0.652741	0.629612	0.625522	0.657010

Langkah pertama yang dilakukan untuk mendapatkan matriks *Moore-Penrose Generalized Inverse* adalah dengan mengalikan matriks *H transpose* dengan matriks *H*. Berikut adalah contoh perkalian tersebut pada indeks 1,1:

$$(H^T * H)_{1,1} = ((0.244776 * 0.244776) + (0.226035 * 0.226035) + \dots (0.229701 * 0.229701)) = 0.47291$$

Tabel 4.12 menunjukkan matriks hasil perkalian matriks *H transpose* dengan matriks *H*.

Tabel 4.12 Matriks $H^T H$

$H^T H$	1	2	3
1	0.472909601	0.503472251	1.125488737
2	0.503472251	0.536436917	1.196596607
3	1.125488737	1.196596607	2.70942132

Langkah berikutnya adalah dengan melakukan *inverse* pada matriks yang ditunjukkan pada Tabel 4.12. Hasil *inverse* dari matriks tersebut dapat dilihat pada Tabel 4.13.

Tabel 4.13 Matriks Inverse

$(H^T * H)^{-1}$	1	2	3
1	4342.852884	-3492.446722	-261.5990743
2	-3492.446722	2934.058111	154.9502352
3	-261.5990743	154.9502352	40.60420031

Matriks inverse tersebut kemudian dikalikan dengan matriks *H transpose* untuk mendapatkan matriks *Moore-Penrose Generalized Inverse* (H^+). Berikut adalah contoh perhitungan untuk mendapatkan nilai H^+ pada indeks 1,1:

$$H^+ = (H^T * H)^{-1} * H^T$$

$$H^+_{1,1} = ((4342.852884 * 0.244777) + (-3492.446722 * 0.253716) + (-261.5990743 * 0.552583)) = 32.38364$$

Tabel 4.14 menunjukkan matriks *Moore-Penrose Generalized Inverse* yang akan digunakan untuk menghitung nilai β .

Tabel 4.14 Matriks H^+

H^+	1	2	3	4	5	6	7
1	32.38364	17.76512	11.33052	-17.38714	-7.85772	17.94745	-46.5187
2	-24.8286	-17.8744	-7.24027	20.785468	9.780374	-19.2958	32.33661
3	-2.2828	0.732143	-1.27245	-1.716253	-0.82297	1.29738	5.285058

Nilai β didapatkan dengan menghitung perkalian antara matriks H^+ dengan vektor target. Berikut adalah contoh perhitungan untuk mendapatkan nilai β pada indeks 1:

$$\beta_1 = ((32.38364 * 0.2) + (17.76512 * 0.00000) + \dots + (-46.5187 * 0.3)) = -26.5480872$$

Tabel 4.15 Menunjukkan nilai β secara keseluruhan yang akan digunakan pada proses pengujian.

Tabel 4.15 Nilai Beta estimasi

β
-26.5480872
14.8973054
5.0591182

4.3.2 Perhitungan Manual Proses Pengujian

Langka pertama yang dilakukan pada proses pengujian adalah dengan menghitung nilai matriks inialisasi yang dihitung menggunakan bobot pada Tabel 4.7 dan Tabel 4.8. Tabel 4.16 menunjukkan matriks inialisasi untuk data uji.

Tabel 4.16 Matriks inialisasi data uji

Hinit Uji	1	2	3
1	-1.425	-1.38167	0.465
2	-1.1725	-0.88639	1.326389
3	-1.88833	-1.805	0.906111

Langkah berikutnya adalah dengan mengaktifasi tiap nilai pada matriks inialisasi tersebut menggunakan fungsi aktivasi sigmoid. Matriks hasil aktivasi tersebut dapat dilihat pada Tabel 4.17.

Tabel 4.17 Matriks aktivasi data uji

H(x) uji	1	2	3
----------	---	---	---

1	0.193879	0.2007415	0.6142
2	0.236403	0.2918556	0.790243
3	0.131435	0.1412435	0.712204

Hasil peramalan didapatkan dengan menggunakan Persamaan 2.6, yaitu dengan mengalikan matriks hasil aktivasi dengan nilai β yang didapatkan pada tahap pelatihan. Tabel 4.18 menunjukkan vektor hasil peramalan yang didapatkan.

Tabel 4.18 Vektor hasil peramalan

Peramalan
0.9507005
2.0697352
2.2179328

4.3.3 Proses Denormalisasi

Proses denormalisasi dilakukan untuk mendapatkan nilai asli dari hasil peramalan. Proses tersebut dapat dilakukan dengan menggunakan Persamaan 2.7. Berikut adalah contoh denormalisasi untuk hasil peramalan pada indeks pertama:

$$\text{Denormalisasi}_1 = 0.9507005(7675 - 7425) + 7425 = 7662.675$$

Tabel 4.19 menunjukkan hasil denormalisasi secara keseluruhan.

Tabel 4.19 Vektor denormalisasi

Denormalisasi
7662.675
7942.434
7979.483

4.3.4 Perhitungan Nilai MAPE

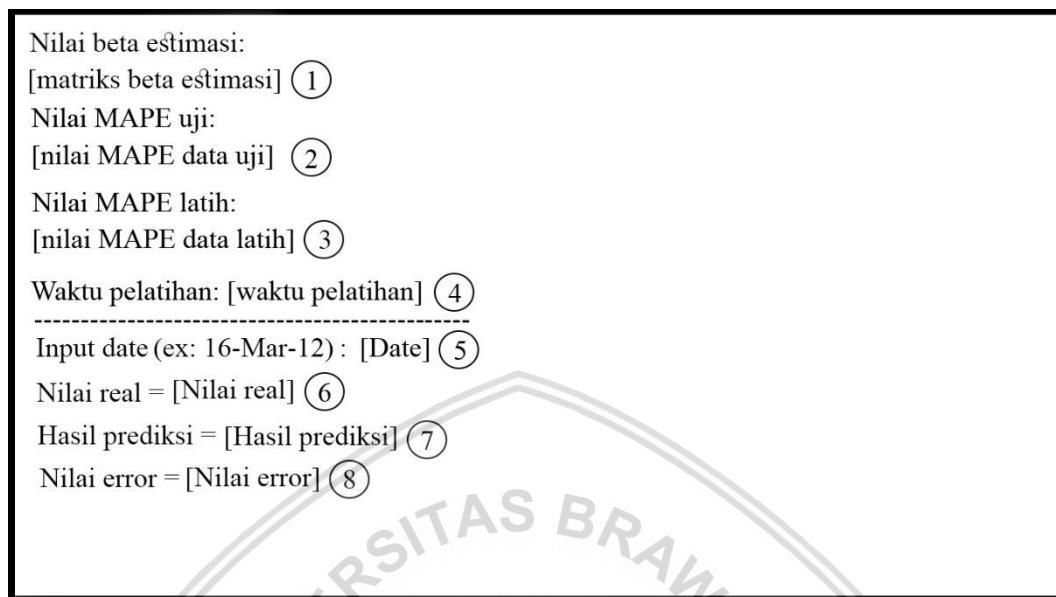
Setelah hasil peramalan didapatkan, perlu dilakukan evaluasi untuk melihat tingkat kesalahan yang didapatkan pada percobaan ini. Evaluasi tersebut dapat dilakukan menggunakan Persamaan 2.8. Berikut adalah hasil perhitungan nilai MAPE dari percobaan ini:

$$\text{MAPE} = \frac{1}{3} \left(\frac{|8000 - 7662.675|}{8000} \right) + \left(\frac{|7950 - 7942.483|}{7950} \right) + \left(\frac{|8000 - 7979.483|}{8000} \right) \\ * 100 = 1.522731\%$$

4.4 Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk memberikan gambaran *output* yang dihasilkan dari implementasi algoritme *Extreme Learning Machine* (ELM) untuk peramalan saham harian Bank Mandiri. Pada penelitian ini, *output* yang dihasilkan

oleh program tidak berbentuk *Graphical User Interface* (GUI), melainkan berupa *Command Line Interface* (CLI). Perancangan antarmuka pada penelitian ini dapat dilihat pada Gambar 4.14.



Nilai beta estimasi:
[matriks beta estimasi] ①
Nilai MAPE uji:
[nilai MAPE data uji] ②
Nilai MAPE latih:
[nilai MAPE data latih] ③
Waktu pelatihan: [waktu pelatihan] ④

Input date (ex: 16-Mar-12): [Date] ⑤
Nilai real = [Nilai real] ⑥
Hasil prediksi = [Hasil prediksi] ⑦
Nilai error = [Nilai error] ⑧

Gambar 4.14 Perancangan antarmuka

Penjelasan lebih lanjut dari Gambar 4.14 adalah sebagai berikut:

1. Matriks nilai β yang didapatkan dari tahap pelatihan.
2. Nilai MAPE data uji.
3. Nilai MAPE data latih.
4. Lama waktu pelatihan algoritme.
5. Input tanggal yang ingin diuji secara manual
6. Nilai sebenarnya pada tanggal yang dimasukkan
7. Hasil prediksi
8. Nilai *error* dari prediksi yang sudah dibuat

4.5 Pengujian Metode

Pengujian metode dilakukan untuk melihat tingkat kesalahan terkecil dari parameter-parameter pengujian yang sudah ditentukan. Pada penelitian ini, pengujian metode meliputi:

1. Pengujian fungsi aktivasi yang digunakan.
2. Pengujian kombinasi jumlah neuron dan variasi jumlah data yang digunakan.

Penelitian tentang parameter terbaik untuk metode ELM sudah pernah dilakukan sebelumnya dan menghasilkan kesimpulan bahwa jumlah neuron yang lebih banyak belum tentu menghasilkan performa ELM yang lebih baik (Cao, 2017). Karena itu akan dilakukan pengujian untuk mengetahui jumlah neuron dan variasi

jumlah data yang menghasilkan nilai *error* terendah. Sedangkan penggunaan fungsi aktivasi sigmoid cenderung lebih menghasilkan performa ELM yang lebih baik dibanding fungsi aktivasi lainnya.

4.5.1 Pengujian Fungsi Aktivasi

Pengujian ini dilakukan untuk mengetahui fungsi aktivasi mana yang dapat memberikan performa ELM terbaik. Tabel 4.20 menunjukkan perancangan pengujian fungsi aktivasi.

Tabel 4.20 Perancangan pengujian fungsi aktivasi

Fungsi aktivasi	Nilai MAPE pada percobaan ke-i										
	1	2	3	4	5	6	7	8	9	10	Rata-rata (%)
Sigmoid											
Sin											
Tanh											

4.5.2 Pengujian Kombinasi Jumlah Neuron dan Variasi Jumlah Data

Pengujian ini dilakukan untuk mengetahui kombinasi jumlah neuron and variasi jumlah data mana yang dapat memberikan hasil terbaik. Tabel 4.21 menunjukkan perancangan pengujian kombinasi jumlah neuron dan variasi jumlah data.

Tabel 4.21 Perancangan pengujian kombinasi jumlah neuron dan variasi jumlah data

Jumlah neuron	Nilai MAPE pada percobaan ke-i										
	1	2	3	4	5	6	7	8	9	10	Rata-rata (%)
1											
2											
3											
4											
5											
10											
25											
50											

4.6 Implementasi Algoritme

4.6.1 Implementasi Normalisasi Data

```

1  def fetchData(rows):
2      global file_path
3      global trainSize
4      df = pd.read_csv(file_path, nrows=rows)
5      openCol = df['Open'].tolist()
6      highCol = df['High'].tolist()
7      lowCol = df['Low'].tolist()
8      trainOpenCol = openCol[:int(trainSize*rows)]
9      trainHighCol = highCol[:int(trainSize*rows)]
10     trainLowCol = lowCol[:int(trainSize*rows)]
11     normOpenCol = []
12     normHighCol = []
13     normLowCol = []
14     for x in range(0, len(openCol)):
15         normOpenCol.append((float(openCol[x]) -
16 min(trainOpenCol)) / (max(trainOpenCol) -
17 min(trainOpenCol)))
18         normHighCol.append((float(highCol[x]) -
19 min(trainHighCol)) / (max(trainHighCol) -
20 min(trainHighCol)))
21         normLowCol.append((float(lowCol[x]) -
22 min(trainLowCol)) / (max(trainLowCol) - min(trainLowCol)))
23
24     mergelist = [list(l) for l in zip(normOpenCol,
25 normHighCol, normLowCol)]
26     return mergelist

```

Gambar 4.15 Kode program normalisasi data

Penjelasan dari kode program pada Gambar 4.15 adalah sebagai berikut:

Baris 2: Definisi variabel global `file_path`

Baris 3: Definisi variabel global `trainSize`

Baris 4: Proses untuk pengambilan data dari file csv dengan menggunakan library pandas

Baris 5-7: Proses untuk mengambil data pada fitur *Open*, *High* dan *Low* kemudian mengkonversi data tersebut menjadi bentuk list.

Baris 8-10: Proses untuk mendapatkan data latih fitur *Open*, *High* dan *Low*.

Baris 11-13: Pembentukan list kosong yang nantinya akan diisi data yang dinormalisasi

Baris 14-22: Proses normalisasi untuk masing-masing fitur dengan menggunakan *Min-Max Normalization*

Baris 24-25: Menggabungkan list fitur *Open*, *High*, dan *Low* menjadi satu buah list

Baris 26: Mengembalikan nilai berupa list gabungan fitur *Open*, *High* dan *Low*

4.6.2 Implementasi Pembangkitan Bobot Input dan Bobot Bias

```

1  def generateBobotInput(a, b):

```


2	matrix = [[float("{0:.2f}".format(random.uniform(-1,
3	1))) for x in range(a)]
4	for y in range(b)]
5	
6	return matrix

Gambar 4.16 Kode program pembangkitan bobot input

Penjelasan dari kode program pada Gambar 4.16 adalah sebagai berikut:

Baris 2-4: Proses untuk membangkitkan nilai random dengan range -1 hingga 1

Baris 6: Mengembalikan nilai berupa matriks bobot input

1	def generateBias(i):
2	bias = [[float("{0:.2f}".format(random.uniform(-1, 1)))
3	for x in range(1)]
4	for y in range(i)]
5	
6	return bias

Gambar 4.17 Kode program pembangkitan bobot bias

Penjelasan dari kode program pada Gambar 4.17 adalah sebagai berikut:

Baris 2-4: Proses untuk membangkitkan nilai random dengan range -1 hingga 1

Baris 6: Mengembalikan nilai berupa matriks bobot bias

4.6.3 Implementasi Pembangkitan Matriks Inisialisasi

1	def generateMatrixInisialisasi(x, w, b):
2	transpose = [[w[j][i] for j in range(len(w))] for i in
3	range(len(w[0]))]
4	
5	result = [[sum(a*b for a,b in zip(x_row,transpose_col))
6	for transpose_col in zip(*transpose)]
7	for x_row in x]
8	
9	for i in range(0, len(result)):
10	for j in range(0, len(result[0])):
11	result[i][j] = result[i][j] + b[j][0]
12	
13	return result

Gambar 4.18 Kode program pembangkitan matriks inisialisasi

Penjelasan dari kode program pada Gambar 4.18 adalah sebagai berikut:

Baris 2-3: Proses untuk melakukan transpose terhadap matriks w (matriks bobot input)

Baris 5-7: Proses untuk mengalikan matriks input dengan matriks bobot input

Baris 9-11: Proses untuk menambahkan bias terhadap matriks hasil perkalian matriks input dengan matriks bobot input

Baris 13: Mengembalikan nilai berupa matriks inisialisasi

4.6.4 Implementasi Fungsi Aktivasi

1	def fungsiAktivasi(H):
2	for i in range(len(H)):

3	for j in range(len(H[0])):
4	H[i][j] = 1 / (1 + math.exp(H[i][j] * (-1)))
5	return H

Gambar 4.19 Kode program fungsi aktivasi

Penjelasan dari kode program pada Gambar 4.19 adalah sebagai berikut:

Baris 2-4: Proses untuk melakukan aktivasi terhadap matriks inisialisasi

Baris 5: Mengembalikan nilai berupa matriks yang sudah diaktivasi

4.6.5 Implementasi *Pseudo-Inverse*

1	def pseudoInverse(H):
2	H = np.matrix(H)
3	Ht = H.getT()
4	preInv = Ht * H
5	Hinv = preInv.getI()
6	Hplus = Hinv * Ht
7	Hplus = Hplus.tolist()
8	return Hplus

Gambar 4.20 Kode program *Pseudo-Inverse*

Penjelasan dari kode program pada Gambar 4.20 adalah sebagai berikut:

Baris 2: Proses untuk mengkonversi matriks hasil aktivasi yang berbentuk list menjadi objek matriks numpy

Baris 3: Proses untuk mendapatkan matriks transpose dari matriks H

Baris 4: Proses untuk perkalian matriks H dan matriks H transpose

Baris 5: Proses untuk mendapatkan inverse dari matriks hasil perkalian matriks H dan matriks H transpose

Baris 6: Proses untuk mengalikan matriks inverse dan matriks H transpose

Baris 7: Proses untuk mengkonversi matriks Hplus menjadi tipe list

Baris 8: Mengembalikan nilai berupa matriks *Pseudo-Inverse*

4.6.6 Implementasi Menghitung Nilai β

1	def betaEstimasi(h, y):
2	Beta = [[sum(a*b for a,b in zip(h_row, y_col)) for
3	y_col in zip(*y)]
4	for h_row in h]
5	
6	return Beta

Gambar 4.21 Kode program menghitung nilai β

Penjelasan dari kode program pada Gambar 4.21 adalah sebagai berikut:

Baris 2-4: Proses untuk mengalikan matriks *Pseudo-Inverse* dengan matriks target

Baris 6: Mengembalikan nilai berupa matriks nilai β

4.6.7 Implementasi Proses Menghitung Hasil Peramalan

1	def hitungPrediksi(h, beta):
---	------------------------------

2	Ypred = [[sum(a*b for a,b in zip(h_row, beta_col)) for
3	beta_col in zip(*beta)]
4	for h_row in h]
5	return Ypred

Gambar 4.22 Kode program proses menghitung hasil peramalan

Penjelasan dari kode program pada Gambar 4.22 adalah sebagai berikut:

Baris 2-4: Proses perkalian antara matriks aktivasi data uji dengan matriks nilai β

Baris 5: Mengembalikan nilai berupa matriks hasil peramalan

4.6.8 Implementasi Denormalisasi Hasil Peramalan

1	def denormalisasi(pred, rows):
2	global file_path
3	global trainSize
4	df = pd.read_csv(file_path, nrows=rows)
5	priceCol = df['Price'].tolist()
6	trainPriceCol = priceCol[:int(trainSize*rows)]
7	maxval = max(trainPriceCol)
8	minval = min(trainPriceCol)
9	denorm = []
10	for x in range(0, len(pred)):
11	denorm.append(pred[x][0] * (maxval-minval) +
12	minval)
13	return denorm

Gambar 4.23 Kode program denormalisasi hasil peramalan

Penjelasan dari kode program pada Gambar 4.23 adalah sebagai berikut:

Baris 2: Definisi variabel global file_path

Baris 3: Definisi variabel global trainSize

Baris 4: Proses untuk mendapatkan data dalam bentuk csv menggunakan library pandas

Baris 5: Proses untuk mendapatkan data pada fitur *Price* dan mengkonversinya menjadi bentuk list

Baris 6: Proses untuk mendapatkan data latih dari fitur *Price*

Baris 7-8: Proses untuk mendapatkan nilai maksimum dan minimum dari fitur *Price*

Baris 9: Pembentukan list kosong yang nantinya akan diisi data hasil denormalisasi

Baris 11-12: Proses denormalisasi

Baris 13: Mengembalikan nilai berupa matriks hasil peramalan yang sudah didenormalisasi

4.6.9 Implementasi Evaluasi Hasil Peramalan

1	def hitungMape(y, ypred):
2	Yabs = 0
3	for i in range(0, (len(y))):
4	if(y[i] == 0):
5	Yabs = Yabs + 0
6	else:

7	Yabs = Yabs + ((math.fabs(y[i] - ypred[i]))/
8	y[i])
9	MAPE = (Yabs / len(y)) * 100
10	return MAPE

Gambar 4.24 Kode program evaluasi hasil peramalan

Penjelasan dari kode program pada Gambar 4.24 adalah sebagai berikut:

Baris 2: Inisialisasi variable Yabs dengan nilai 0

Baris 3-9: Proses evaluasi hasil peramalan dengan menghitung nilai MAPE

Baris 10: Mengembalikan nilai berupa hasil MAPE

4.7 Implementasi Antarmuka

Implementasi dari perancangan antarmuka yang berbentuk *Command Line Interface* (CLI) pada bab perancangan dapat dilihat pada Gambar 4.25.

```

Nilai beta estimasi:
[1.1224899358043761]
[1.0491723007908269]
[-9.9165019089174073]
[11.304822389913781]

Nilai MAPE uji:
1.02736622277 %

Nilai MAPE latih:
1.00007899793 %

Waktu pelatihan: 0.001999855s
-----

Input date (ex: 16-Mar-12): 20-Oct-17

Nilai real = 6950
Hasil prediksi = 7042.0
Nilai error = 1.32374100719%
  
```

Gambar 4.25 Gambar implementasi antarmuka

BAB 5 PENGUJIAN DAN ANALISIS

Bab ini akan membahas tentang pengujian dan analisis hasil peramalan saham harian Bank Mandiri menggunakan metode *Extreme Learning Machine*. Tujuan dari pengujian dan analisis ini adalah untuk mengetahui parameter seperti apa yang menghasilkan nilai *error* terkecil.

5.1 Pengujian Fungsi Aktivasi

Tabel 5.1 memperlihatkan hasil pengujian dengan menggunakan fungsi aktivasi sigmoid, sin dan tanh.

Tabel 5.1 Pengujian fungsi aktivasi sigmoid, sin dan tanh

Fungsi aktivasi	Nilai MAPE pada percobaan ke-i										
	1	2	3	4	5	6	7	8	9	10	Rata-rata (%)
Sigmoid	1,28	1,27	1,27	1,29	1,27	1,27	1,29	1,26	1,31	1,29	1,28
Sin	1,35	1,37	1,31	1,30	1,37	1,30	1,29	1,30	1,28	1,29	1,316
Tanh	1,44	1,27	1,27	1,32	1,35	1,30	1,28	1,34	1,30	1,35	1,322

Berdasarkan Tabel 5.1, dapat dilihat bahwa fungsi aktivasi sigmoid memberikan hasil terbaik dalam bentuk nilai MAPE yang rendah. Pengujian ini memperkuat pernyataan bahwa fungsi aktivasi sigmoid merupakan fungsi terbaik untuk diimplementasikan pada kasus regresi (Cao, 2017).

5.2 Pengujian Kombinasi Jumlah Neuron dan Variasi Jumlah Data

Tabel 5.2, Tabel 5.3 dan Tabel 5.4 menunjukkan hasil pengujian kombinasi jumlah neuron dan variasi jumlah data yang digunakan.

Tabel 5.2 Pengujian jumlah neuron pada data 5 tahun terakhir

Jumlah neuron	Nilai MAPE pada percobaan ke-i										
	1	2	3	4	5	6	7	8	9	10	Rata-rata (%)
1	46,44	56,07	42,73	53,15	57,24	39,58	54,52	47,03	40,64	45,49	48,289
2	3,54	3,15	3,14	1,62	4,91	1,53	2,25	2,87	6,68	1,66	3,135
3	1,34	1,41	2,15	1,29	2,22	1,26	1,78	2,07	1,43	1,59	1,654
4	2,50	1,39	1,96	1,43	1,51	1,26	1,93	1,33	1,32	2,84	1,747
5	1,28	1,60	1,51	1,34	1,26	1,29	1,25	1,25	1,27	1,28	1,333

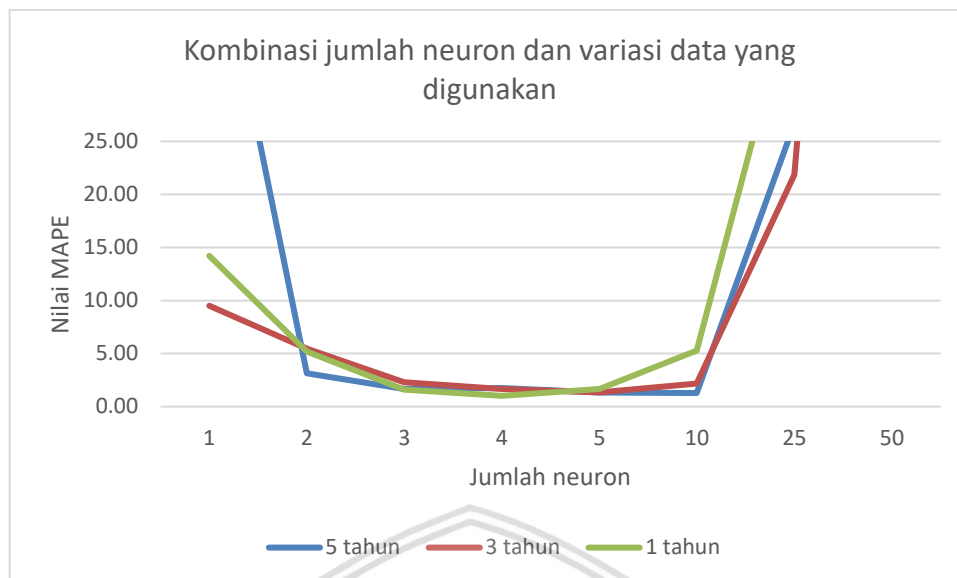
10	1,28	1,27	1,27	1,29	1,27	1,27	1,29	1,26	1,31	1,29	1,28
25	29,22	12,98	60,88	18,74	31,77	6,53	32,01	24,68	68,56	11,40	26,677
50	82,74	128,5	8,22	101,1	127,2	170,2	62,7	735,7	373,1	249,2	203,86

Tabel 5.3 Pengujian jumlah neuron pada data 3 tahun terakhir

Jumlah neuron	Nilai MAPE pada percobaan ke-i										
	1	2	3	4	5	6	7	8	9	10	Rata-rata (%)
1	8,25	11,61	9,61	8,30	9,24	8,98	9,08	10,27	10,26	9,61	9,521
2	3,64	8,16	7,01	1,28	8,20	13,76	3,19	4,49	2,91	2,11	5,475
3	1,96	1,85	1,30	3,89	1,21	1,68	0,97	1,41	7,39	1,42	2,308
4	1,48	1,06	1,53	2,13	1,05	1,72	1,56	2,39	1,39	2,36	1,667
5	1,40	1,13	1,40	1,65	1,39	1,16	1,36	1,32	1,29	1,30	1,34
10	1,87	2,07	2,23	2,34	2,68	2,49	1,88	2,66	1,78	1,84	2,184
25	34,62	5,07	16,70	23,02	35,77	4,44	34,03	29,50	20,62	14,73	21,85
50	93,94	155,7	69,58	48,20	172,8	163,5	413,3	109,1	42,04	31,9	130,06

Tabel 5.4 Pengujian jumlah neuron pada data 1 tahun terakhir

Jumlah neuron	Nilai MAPE pada percobaan ke-i										
	1	2	3	4	5	6	7	8	9	10	Rata-rata (%)
1	14,95	18,13	12,88	12,15	13,20	15,90	13,19	13,70	14,38	13,62	14,21
2	1,17	6,65	7,91	1,62	12,95	1,60	13,51	1,16	1,31	4,04	5,192
3	1,74	0,93	1,06	2,73	1,88	1,21	1,93	1,85	1,11	1,61	1,605
4	1,08	0,96	0,93	0,97	0,89	0,91	1,36	1,00	1,08	0,98	1,016
5	1,69	1,53	1,78	1,73	0,91	1,92	1,44	2,29	1,43	1,93	1,665
10	6,04	10,86	2,16	3,61	4,21	5,26	7,61	4,87	4,40	3,86	5,288
25	63,28	56,82	28,70	100,7	17,12	8,81	53,11	9,82	29,13	31,77	39,926
50	400,5	144,8	79,36	38,14	28,18	34,66	124,2	62,57	38,22	182,1	113,27



Gambar 5.1 Kombinasi jumlah neuron dan variasi jumlah data yang digunakan

Gambar 5.1 menunjukkan grafik nilai MAPE dari kombinasi – kombinasi yang ada pada Tabel 5.2, Tabel 5.3 dan Tabel 5.4. Hasilnya, kombinasi yang menghasilkan nilai MAPE terendah adalah penggunaan neuron sebanyak empat buah dan menggunakan data satu tahun terakhir.

5.3 Pengujian Dengan Data Uji yang Sama Pada Setiap Skenario

Pengujian ini dilakukan menggunakan data uji sebanyak satu bulan terakhir. Tabel 5.5 menunjukkan hasil pengujian yang sudah dilakukan.

Tabel 5.5 Pengujian dengan data uji yang sama pada setiap skenario

Banyak data latih	Nilai MAPE pada percobaan ke-i										
	1	2	3	4	5	6	7	8	9	10	Rata-rata (%)
1 tahun	1,16	1,04	0,94	1,08	0,93	0,93	1,05	1,00	1,03	0,96	1,012
3 tahun	1,59	1,11	1,16	1,1	1,11	1,23	1,41	3,75	1,37	1,12	1,495
5 tahun	1,24	1,00	1,02	1,5	1,7	1,12	0,97	1,15	1,21	1,29	1,22

Dari Tabel 5.5 dapat dilihat bahwa penggunaan data latih sebanyak satu tahun terakhir tetap menghasilkan nilai MAPE yang lebih rendah dibanding penggunaan data latih sebanyak tiga tahun dan lima tahun.

Gambar 5.2 menunjukkan grafik perbandingan nilai saham sebenarnya dan nilai hasil peramalan menggunakan metode ELM pada saat nilai *error* sebesar 0.95%.



Gambar 5.2 Grafik perbandingan nilai real dan nilai hasil peramalan

5.4 Analisis Hasil Pengujian

5.4.1 Analisis Hasil Pengujian Fungsi Aktivasi

Pada penelitiannya, Cao mengatakan bahwa fungsi aktivasi sigmoid adalah fungsi aktivasi yang terbaik untuk permasalahan berbentuk regresi. Pengujian ini membuktikan bahwa fungsi aktivasi sigmoid menghasilkan performa yang paling baik diantara ketiga fungsi aktivasi yang diuji. Namun apabila melihat hasil terbaik dari masing-masing fungsi aktivasi, hasil dari fungsi aktivasi sigmoid hanya memiliki selisih yang kecil dari dua fungsi aktivasi lainnya. Hal ini dapat terjadi karena rentang nilai pemetaan ketiga fungsi aktivasi tersebut mirip. Pemetaan dari fungsi aktivasi sigmoid berada pada rentang nilai 0,1, sedangkan fungsi aktivasi sin dan tanh berada pada rentang nilai -1,1.

Dari analisis tersebut, dapat disimpulkan bahwa penggunaan fungsi aktivasi sigmoid dapat menghasilkan performa ELM yang lebih baik.

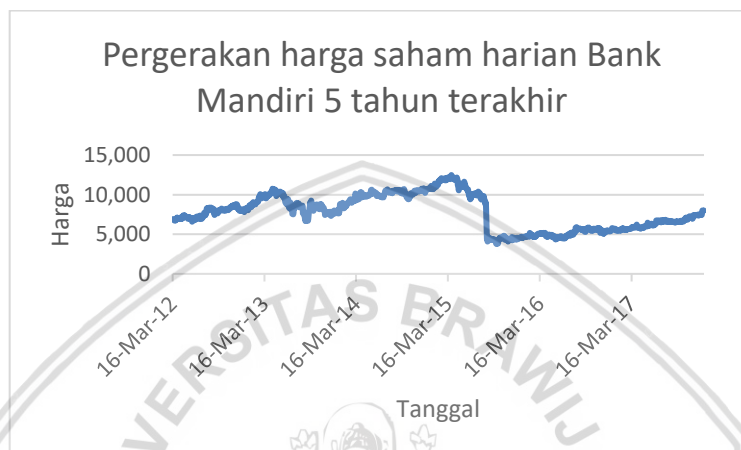
5.4.2 Analisis Hasil Pengujian Kombinasi Jumlah Neuron dan Variasi Jumlah Data

Dari hasil pengujian yang sudah dilakukan, didapatkan kombinasi yang menghasilkan nilai MAPE terendah adalah menggunakan empat buah neuron dan menggunakan data saham harian Bank Mandiri satu tahun terakhir.

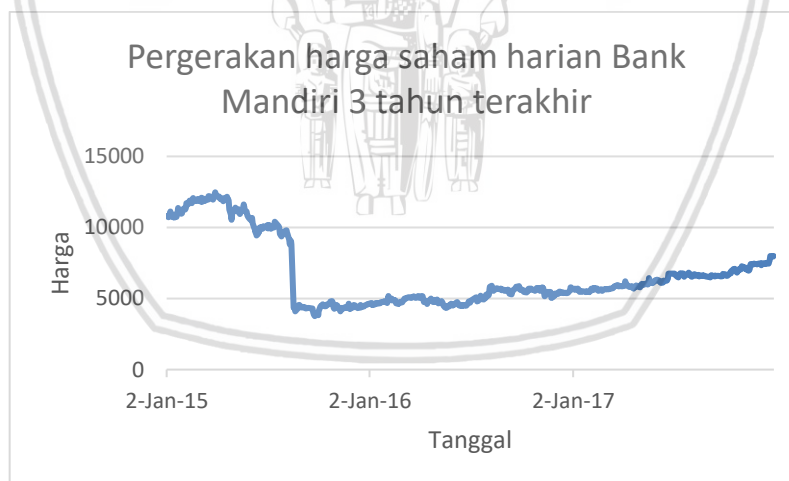
Grafik nilai MAPE yang ada pada Gambar 5.1 juga menunjukkan garis yang membentuk seperti kurva dan hal ini menunjukkan bahwa penggunaan neuron yang lebih sedikit atau lebih banyak tidak berarti akan menghasilkan performa ELM yang lebih baik. Penggunaan jumlah neuron yang terlalu banyak juga dapat menimbulkan masalah *overfitting* dan menghasilkan performa generalisasi yang lebih buruk (Ke, Liu, 2008). Dari Gambar 5.1, dapat dilihat bahwa penggunaan neuron di atas sepuluh buah akan menghasilkan nilai MAPE yang tinggi. Hal ini

disebabkan karena data latih yang digunakan pada penelitian ini memiliki jarak yang berdekatan sehingga masalah *overfitting* dapat timbul lebih cepat.

Penggunaan data selama satu tahun terakhir juga menghasilkan performa ELM yang lebih baik dibanding penggunaan data selama tiga tahun dan lima tahun terakhir. Hal ini dapat disebabkan karena fluktuasi harga saham harian Bank Mandiri selama satu tahun terakhir lebih stabil dibandingkan data tiga tahun dan lima tahun terakhir seperti ditunjukkan pada Gambar 5.3, Gambar 5.4 dan Gambar 5.5.



Gambar 5.3 Pergerakan harga saham harian Bank Mandiri 5 tahun terakhir



Gambar 5.4 Pergerakan harga saham harian Bank Mandiri 3 tahun terakhir



Gambar 5.5 Pergerakan harga saham harian Bank Mandiri 1 tahun terakhir

Pada Gambar 5.3 dan 5.4 terlihat ada titik dimana harga saham harian Bank Mandiri mengalami fluktuasi yang tinggi, sedangkan pada Gambar 5.5 pergerakan harga saham harian Bank Mandiri stabil. Hal ini yang kemudian menyebabkan mengapa penggunaan data satu tahun terakhir menghasilkan performa yang lebih baik dibanding penggunaan data tiga tahun dan lima tahun terakhir.

5.4.3 Analisis Hasil Pengujian Dengan Data Uji yang Sama Pada Setiap Skenario

Dari hasil pengujian yang sudah dilakukan, sekali lagi terbukti bahwa penggunaan data latih sebanyak satu tahun terakhir menghasilkan nilai *error* yang lebih rendah dibanding data tiga tahun dan lima tahun terakhir. Sama seperti hasil analisis pada pengujian sebelumnya, hal ini disebabkan oleh fluktuasi harga saham yang lebih stabil pada penggunaan data latih satu tahun terakhir sebagaimana dapat dilihat pada Gambar 5.5. Sedangkan pada data tiga tahun dan lima tahun terakhir terdapat fluktuasi harga yang tinggi sebagaimana dapat dilihat pada Gambar 5.3 dan Gambar 5.4.

5.3 Cross Validation Data Time Series

Cross Validation dilakukan untuk mengevaluasi performa model yang sudah dibuat. Pada penelitian ini akan dilakukan *11-fold cross validation* pada data saham harian Bank Mandiri satu tahun terakhir. Tabel 5.6 menunjukkan hasil *cross validation* yang sudah dilakukan.

Fold train menyatakan bulan yang digunakan untuk data latih sedangkan *fold test* menyatakan bulan yang digunakan untuk data uji. Contohnya, pada *fold* pertama, yang digunakan sebagai data latih adalah data bulan Januari, sedangkan yang digunakan sebagai data uji adalah data bulan Februari. Pada setiap *fold* dilakukan percobaan sebanyak lima kali dan kemudian diambil rata-rata nilai MAPEnya.

Tabel 5.6 Tabel Cross Validation

<i>Fold train</i>	<i>Fold test</i>	T1	T2	T3	T4	T5	Rata-rata MAPE (%)
1	2	1,02	1,02	1,03	1,02	0,98	1,014
1-2	3	1,16	1,06	1,36	1,06	1,23	1,174
1-3	4	1,35	1,35	1,35	1,41	1,32	1,356
1-4	5	1,55	1,69	1,49	1,54	1,51	1,556
1-5	6	1,35	1,35	1,33	1,34	1,36	1,346
1-6	7	1,38	1,39	1,36	1,32	1,29	1,348
1-7	8	0,94	0,77	1,03	0,58	0,85	0,834
1-8	9	0,56	0,66	0,56	0,55	0,6	0,586
1-9	10	1,04	0,93	0,96	1,14	0,89	0,992
1-10	11	1,09	0,94	1,01	1	1,44	1,096
1-11	12	1,11	1,32	1,02	1,18	0,95	1,116
Rata-rata							1,128

Hasil dari validasi yang sudah dilakukan menunjukkan performa ELM pada data saham harian Bank Mandiri satu tahun terakhir. Setiap *fold* tidak menunjukkan adanya tanda-tanda kasus *underfitting* ataupun *overfitting*. Rata-rata nilai MAPE dari *cross validation* yang dilakukan adalah sebesar 1.128%.

BAB 6 KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan pengujian dan analisis yang dilakukan, dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Nilai MAPE terbaik yang dihasilkan untuk peramalan harga saham harian Bank Mandiri dengan menggunakan metode *Extreme Learning Machine* adalah sebesar 1,012%.
2. Kombinasi parameter yang menghasilkan nilai MAPE terbaik adalah penggunaan fungsi aktivasi sigmoid, jumlah neuron sebanyak empat buah, data yang digunakan sebagai data latih adalah data satu tahun terakhir dan data uji yang digunakan adalah data satu bulan terakhir. Apabila neuron yang digunakan berjumlah kurang dari empat buah, maka akan ada kemungkinan *underfitting*, sedangkan apabila neuron yang digunakan berjumlah lebih dari sepuluh buah, maka akan ada kemungkinan *overfitting*. Hasil *cross validation* yang telah dilakukan juga tidak menunjukkan adanya tanda-tanda *underfitting* ataupun *overfitting* pada model yang dibuat.

6.2 Saran

Dari kesimpulan yang diperoleh, ada beberapa saran yang dapat diberikan untuk pengembangan penelitian ini, diantaranya:

1. Mencari fitur-fitur lain yang sekiranya dapat mempengaruhi pergerakan harga saham selain harga *Open*, *High* dan *Low*.
2. Melakukan optimasi terhadap *hyperparameter* yang berpengaruh menggunakan teknik optimasi seperti *Grid Search* untuk mendapatkan model ELM terbaik.

DAFTAR PUSTAKA

- Barus, A. C., Christina, 2014. Pengaruh Reaksi Pasar Terhadap Harga Saham Perusahaan yang Terdaftar di Bursa Efek Indonesia. *Jurnal Wira Ekonomi Mikroskil*, Vol. 4, No. 1.
- Cao, W., et al. 2017. Some Trick in Parameter Selection for Extreme Learning Machine. *IOP Conference Series: Materials Science and Engineering*.
- Cheng, G. J., Cai, L. and Pan, H. X. 2009. Comparison of Extreme Learning Machine with Support Vector Regression for Reservoir Permeability Prediction. *2009 International Conference on Computational Intelligence and Security*, Beijing, pp. 173-176.
- de Chazal, P., Tapson, J. and van Schaik, A. 2015. A comparison of extreme learning machines and back-propagation trained feed-forward networks processing the mnist database. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, QLD, 2015, pp. 2165-2168.
- Fonseca-Delgado, R., Gomez-Gil, P. 2013. An Assessment of Ten-Fold and Monte Carlo Cross Validations for Time Series Forecasting. *2013 10th International Conference on Electrical Engineering, Computer Science and Automatic Control (CCE)*, Mexico City, pp. 215-220.
- Franco, E. F., Ohishi, T. and Salgado, R. M. 2017. Very short-term bus reactive load forecasting models based on KDD approach. *IEEE 7th International Conference on Power and Energy Systems (ICPES)*, Toronto, ON, 2017, pp. 34-41.
- Giusti, A., 2017. Prediksi Penjualan Mi Menggunakan Metode Extreme Learning Machine (ELM) di Kober Mie Setan Cabang Soekarno Hatta. S1. Universitas Brawijaya.
- Huang, G., Zhou, H., Ding, X. & Zhang, R., 2012. Extreme Learning Machine for Regression and Multiclass Classification. *IEEE Transactions on systems, man, and cybernetics—PART B: CYBERNETICS*, VOL. 42, NO. 2.
- Ke, J., Liu, X. 2008. Empirical Analysis of Optimal Hidden Neurons in Neural Network Modeling for Stock Prediction. *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, Wuhan, pp. 828-832.
- Lazarevska, E., 2016. Wind speed prediction with extreme learning machine. *IEEE 8th International Conference on Intelligent Systems (IS)*, Sofia, 2016, pp. 154-159.
- Li, G. X. 2016. Application of Extreme Learning Machine Algorithm in the Regression Fitting. *2016 International Conference on Information System and Artificial Intelligence (ISAI)*, Hong Kong, pp. 419-422.
- Prakaulya, V., Sharma, R., Singh, U. and Itare, R. 2017. Railway passenger forecasting using time series decomposition model. *2017 International*

- conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, pp. 554-558.
- Tee, J. E. C., et al. 2017. Day-ahead forecasting of wholesale electricity pricing using extreme learning machine. *TENCON 2017 - 2017 IEEE Region 10 Conference*, Penang , pp. 2973-2977.
- Wang, Z. M. and Wu, C. 2010. Application of Support Vector Regression Method in Stock Market Forecasting. *2010 International Conference on Management and Service Science*, Wuhan, pp. 1-4.
- Ye, T. 2017. Stock forecasting method based on wavelet analysis and ARIMA-SVR model. *2017 3rd International Conference on Information Management (ICIM)*, Chengdu, pp. 102-106.
- You, Z., et al. 2014. Large-Scale Protein-Protein Interactions Detection by Integrating Big Biosensing Data with Computational Model. *2014 BioMed Research International*.

